'68' MICRO JOURNAL

$2.50 USA

Australia A $ 4.00 New Zealand NZ $ 4.00
Singapore S $ 6.00 Hong Kong H $20.00
Malaysia M $ 8.00

VOLUME III ISSUE I • Devoted to the 68XX User • January 1981

"Small Computers Doing Big Things"

UniFLEX™

Multi-User

UniFLEX is the first full capability multi-user operating system available for microprocessors. Designed for the 6809 and 68000, it offers its users a very friendly computing environment. After a user 'logs-in' with his user name and password, any of the system programs may be run at will. One user may run the text editor while another runs BASIC and still another runs the C compiler. Each user operates in his own system environment, unaware of other user activity. The total number of users is only restricted by the resources and efficiency of the hardware in use.

Multi-Tasking

UniFLEX is a true multi-tasking operating system. Not only may several users run different programs, but one user may run several programs at a time. For example, a compilation of one file could be initiated while simultaneously making changes to another file using the text editor. New tasks are generated in the system by the 'fork' operation. Tasks may be run in the background or 'locked' in main memory to assist critical response times. Inter-task communication is also supported through the 'pipe' mechanism.

Support

The design of UniFLEX, with its hierarchical file system and device independent I/O, allows the creation of a variety of complex support programs. There is currently a wide variety of software available and under development. Included in this list is a Text Processing System for word processing functions, BASIC interpreter and precompiler for general programming and educational use, native C and Pascal compilers for more advanced programming, sort/merge for business applications, and a variety of debug packages. The standard system includes a text editor, assembler, and about forty utility programs. UniFLEX for 6809 is sold with a single CPU license and one years maintenance for $450.00. Additional yearly maintenance is available for $100.00. OEM licenses are also available.

FLEX™

UniFLEX is offered for the advanced microprocessor systems. FLEX, the industry standard for 6800 and 6809 systems, is offered for smaller, single user systems. A full line of FLEX support software and OEM licenses are also available.

tsc technical systems consultants, inc.
Box 2570, West Lafayette, IN 47906
(317) 463-2502 Telex 276143

'68' MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-DMAF2-CDS1-CT82-Sprint 3
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Mainframe-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

Publisher: Don Williams Sr.

Executive Editor: Larry Williams

Subscriptions and Office manager
Mary Robertson

General Girl 'Friday'
Joyce Williams

Contributing Editors:

Dr. Jack Bryant
Dr. Chuck Adams
Dr. Theo Elbert
Dr. Jeffery Brownstein
Dale Puckett
Russell Gore
Ron Anderson
John Jordon
Dennis Womack

* CONTENTS *

Send All Correspondence To:

'68' Micro Journal
3018 Hamill Rd.
PO Box 849
Hixson, Tennessee 37343

— Phone —
Office: 615-870-1993
Plant: 615-892-7544

'68' Micro Journal is published 12 times a year by '68' Micro Journal, 6131 Airways Blvd., Chattanooga, TN 37421. Second Class postage paid at Chattanooga, TN. Postmaster: Send Form 3579 to '68' Micro Journal, PO Box 849, Hixson, TN 37343.

1-Year $18.50 2-Year $32.50 3-Year $48.50

—ITEMS SUBMITTED FOR PUBLICATION—

(Letters to the Editor for Publication) All 'letters to the Editor' should be substantiated by facts. Opinions should be indicated as such. All letters must be signed. We are interested in receiving letters that will benefit or alert our readers. Praise as well as gripes is always good subject matter. Your name may be withheld upon request. If you have had a good experience with a 6800 vendor please put it in a letter. If the experience was bad put that in a letter also. Remember, if you tell us who they are then it is only fair that your name 'not' be withheld. This means that all letters published, of a critical nature, cannot have a name withheld. We will attempt to publish 'verbatim' letters that are composed using 'good taste.' We reserve the right to define (for '68' Micro) what constitutes 'good taste.'

(Articles and items submitted for publication) Please, always include your full name, address, and telephone number. Date and number all sheets. TYPE them if you can, poorly handwritten copy is sometimes the difference between go, no-go. All items should be on 8X11 inch, white paper. Most all art work will be reproduced photographically, this includes all listings, diagrams and other non-text material. All typewritten copy should be done with a NEW RIBBON. All hand drawn art should be black on white paper. Please no hand written code items over 50 bytes. Neatly typed copy will be directly reproduced. Column width should be 3¼ inches.

(Advertising) Any Classified: Maximum 20 words. All single letters and/or numbers will be considered one (1) word. No Commercial or Business Type Classified advertising. Classified ads will be published in our standard format. Classified ads $7.50 one time run, paid in advance.

Commercial and/or Business advertisers please write or phone for current rate sheet and publication lag time.

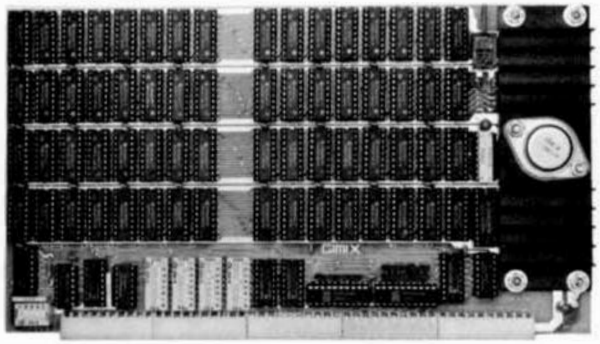
LOOK

WHAT'S COOKING on the FIFTY BUS

32K STATIC RAM BOARDS

Designed for use with:

★ Existing SS50 Systems ★ SS50C Extended Address Systems



- Assembled
- Burned In
- Tested

16K . . . $328.12
24K . . . $438.14
32K . . . $548.15

16K and 24K Versions are socketed for 32K and require only additional 2114's for expansion.

FEATURES:

- Decoding for 4 Extended Address Lines (allows memory decoding up to 1 megabyte)
- DIP-switch to set extended addressing or disable it
- 4 separate 8K blocks, addressable to any 8K boundary by DIP-switch
- Each 8K block may be individually disabled
- Write protect either of two 16K sections
- Low power consumption — uses 2114L low power RAMS
- Fully Socketed
- Gold Bus Connectors
- Guaranteed 2MHz operation

AND NOW...GIMIX OFFERS YOU A Choice of 6800 or 6809 CPU CARDS

You can order your system to fit your needs or select one of the below featured systems. Please contact the factory for further information and availability.

Add as much memory as you need using **GIMIX Static RAM Cards** for the utmost in reliability.

32K 6800 SYSTEM $1,694.59
Includes: Chassis. 6800 CPU, 32K RAM BOARD, I/O card

32K 6809 SYSTEM $1,844.69
Includes: Chassis. 6809 CPU, 32K RAM BOARD, I/O card

32K 6809 PLUS SYSTEM $1,994.79
Includes: Chassis. 32K RAM BOARD. I/O Card, and features our 6809 PLUS CPU Card with the Time of Day Clock option with battery back-up installed, as well as the 6840 Timer Package that provides 3 independent 16 bit counters.

This system also allows the following options to be added at additional cost:

- Battery back-up of the 1K RAM by substituting CMOS parts.
- A 9511 or 9512 Arithmetic Processor.
- GIMIX or SWTP Dynamic Address Translators.

EXPORT NOTES:

For 50Hz 230V C.V. POWER SUPPLY . . . . Add $30.00
80 x 24 VIDEO BOARDS — Specify Format (No Added Charge)

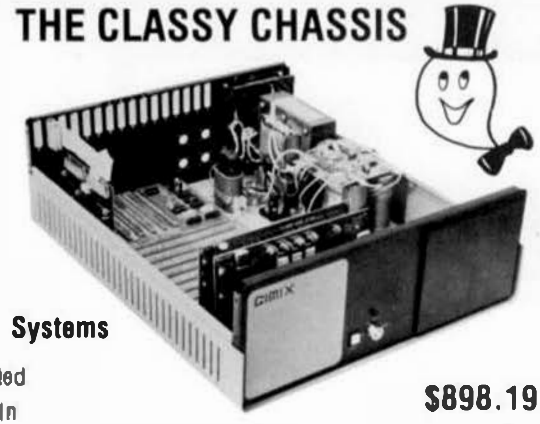
On Orders under $250.00 for a Single Board, or Chips, please Add $10.00 Handling and we will ship Air Mail Prepaid. On all other orders we will ship via Emery Air Freight Collect, and we will charge no handling. All orders must be prepaid in U.S. Funds. Please note that foreign checks have been taking about eight weeks for collection, so we would advise wiring money or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, Account #73-32033. Visa or Master Charge also accepted.

FACTORY PRIME STATIC RAMS

2114L 450 ns . . $5.90 300 ns . . $6.40 200 ns . . $6.90
4044 450 ns . . $5.90 250 ns . . $6.90

ADD $5.00 HANDLING ON ORDERS UNDER $200.00

THE CLASSY CHASSIS



$898.19

- 25 amp (5V) ferro-resonant constant voltage power supply,
- Heavy weight aluminum cabinet with 3 position key switch, fan, and provisions for two 5" disk drives;
- 6800/6809 Mother Board, fifteen 50 pin and eight DIP-switch addressable 30 pin slots (gold plated pins), fully decoded;
- Baud rate generator on I/O section of Mother Board.

I/O BOARDS

for the 30 PIN BUS:

1 Port Serial . . . . (RS 232 or 20MA, current loop)	$ 88.41
2 Port RS 232 Serial	128.43
2 Port Parallel	88.42

for the 50 PIN BUS:

8 Port RS 232 Serial	288.40
8 Port RS 232 Serial . . . . with on board Baud Rate generator.	318.46
8 Port Parallel	198.45

BOTH 6809 SYSTEMS FEATURE OUR NEW TERMINAL BASED GMXBUG 09 SYSTEM MONITOR

GMXBUG 09 includes advanced debugging tools, utility, and memory manipulation routines.

Both 6809 Systems:

★ Can be reconfigured to allow use of other system monitors (OS-9 and SBUG-E)

★ Include 1K of Scratchpad RAM on the CPU

★ Allow optional software switching of system monitors.

2MHz 6809's at slight additional cost when they become available.

Phone, write, or see your dealer for details and prices on our broad range of Boards and Systems for the SS50/SS50C bus and our AC Power Control Products for all computers.

GIMIX inc.

The Company that delivers
Quality Electronic products since 1975.

1337 WEST 37th PLACE, CHICAGO, IL 60609
(312) 927-5510 • TWX 910-221-4055

GIMIX® and GHOST® are Registered Trademarks of GIMIX INC.

See Ghost Ad Pages 36,39,40,48

INNOVATION AND PERFORMANCE

6800 SOFTWARE SUPER POWER

BY MICROWARE®

A/BASIC COMPILER

This BASIC compiler generates pure, fast, efficient 6800 machine language from easy to write BASIC source programs. Uses ultra-fast integer math, extended string functions, boolean operators and real-time operations. Output is ROMable and *runs without any run-time package*. Disk versions have disk I/O statements and require 12K memory and host DOS. Cassette version runs in 8K and requires RT/68 operating system.

☐ Disk Extended Version 2.1 SSB or FLEX* Diskette $150.00

☐ Cassette Version 1.0. $65.00

A/BASIC SOURCE GENERATOR

An "add-on" option for A/BASIC Compiler disk versions that adds an extra third pass which generates a full assembly-language output listing *and* assembly language source file. Uses original BASIC names and inserts BASIC source lines as comments.

☐ SSB or FLEX* Diskette $75.00

A/BASIC INTERPRETER

Here it is - a super-fast A/BASIC compiler! Now you can interactively edit, execute and debug A/BASIC programs with the ease of an interpreter - then compile to super efficient machine language. Also a superb stand-alone applications and control- oriented interpreter. Requires 8K RAM. The cassette version is perfect for Motorola D2 kits.

☐ Cassette, SSB or FLEX* Diskette $95.00

LISP INTERPRETER

The programming Language LISP offers exciting new possibilities for microcomputer applications. A highly interactive interpreter that uses list-type data structures which are simultaneously data and executable instructions. LISP features an unusual structured, recursive function-oriented syntax. Widely used for processing, artificial intelligence, education, simulation symbolic, and computer-aided design. 6800 LISP requires a minimum of 12K RAM.

☐ Cassette, SSB or FLEX* Diskette $75.00

*FLEX is a trademark of Technical Systems Consultants

RT/68 REAL TIME OPERATING SYSTEM

MIKBUG — compatible ROM that combines an improved monitor/debugger with a powerful multitasking real-time operating system. Supports up to 16 concurrent tasks at 8 priority levels plus real time clock and interrupt control. Thousands in use since 1976 handling all types of applications. Available on 6830 (MIKBUG-type) or 2708 (EPROM-type) ROM. Manual is a classic on 6800 real-time applications and contains a full source program listing.

☐ RT68MX (6830) $55.00

☐ RT68MXP (2708) $55.00

6800 CHESS

A challenging chess program for the 6800. Two selectable difficulty levels. Displays formatted chess boared on standard terminals. Requires 8K memory. Machine language with A/BASIC source listing.

☐ Cassette, SSB or FLEX* Diskette $50.00

Our software is available for most 6800 systems on cassette or diskette unless otherwise noted. Phone orders welcomed. We accept MASTERCHARGE and VISA. We try to ship orders within 24 hours of receipt. Please call or write if you require additional information or our free catalog. Microware® software is available for OEM and custom applications.

MICROWARE®

Microware Systems Corporation
P.O. Box 4865, Des Moines, IA 50304
(515) 279-8844

MICROSOFT DISK EXTENDED BASIC FOR OS-9™

Microsoft Basic is by far the world's most popular Basic language — and a vast library of applications software written for it can be used on your system: business packages, scientific, engineering and educational programs, games, etc. The OS-9™ version of Microsoft Basic is a full implementation of Release 5.0 with the following features:

- Four variable types: Integer, String, Single Precision Floating Point (7 Digit), Double Precision Floating Point (16 Digit).
- Full PRINT USING for formatted output (includes asterisk fill, floating $, scientific notation, trailing sign, comma insertion).
- Trace Facilities for program debugging.
- Extensive program editing facilities via EDIT command.
- Matrices with up to 255 dimensions.
- IF/THEN/ELSE and WHILE/WEND for structured programming.
- Automatic Line numbering and renumber.
- Dynamic string space allocation.
- Random and sequential file I/O with variable length records.
- Protected files can be saved in coded binary format.
- CHAIN and COMMON statements — programs may be linked together and share common variables.

This version of Microsoft Basic is not just a reassembled 6800 Basic — it has been enhanced to take full advantage of the 6809 and OS-9™ superior capabilities. It is also a reliable Basic that you can count on for your important programs.

☐ Microsoft Extended Basic Release 5.0 for OS-9™ $250.00

☐ **Also available:** Standard Microsoft 6800 or 6809 Basic Release 4.51 for Flex*. Many features of OS-9™ version. $250.00 *Trademark of Technical Systems Consultants

MICROWARE.

5835 Grand, Box 4865, Des Moines, IA 50304 • (515) 279-8844

HEAR YE!

HEAR YE! HEAR YE!

SOFTWARE ANNOUNCEMENT

NOW INCLUDES LIBJCP

JCP
Job Control Program
By Peter Murray

The JOB CONTROL PROGRAM (JCP) reads a text file that contains the necessary input for a program and then supplies this input to the program in the same manner that an operator would have normally entered it from the keyboard. The file containing the input is referred to as the procedure file, and the program receiving the input is referred to as the calling program. A procedure file contains input for such calling programs as FLEX™, FLEX utility commands, and other development software.

LIBJCP is used as a FLEX™ command within a procedure to load and execute another procedure. At the conclusion of the called procedure, control will return to the calling procedure and execution will resume at the line following the LIBJCP command.

JCP also provides for parameter substitution within the procedure file, special commands to control JCP program flow, and a means for recovery from processing errors. These features allow for commonly used file routines to be written as a generalized procedure that JCP will execute, unattended, simply by entering a single FLEX command.

PRICE $89.95

See Review in July '80 '68' Micro.

AVAILABLE NOW!

REMOTE
Intelligent Terminal Program
By Tom Speer

REMOTE allows use of your 6800 or 6809 system as a terminal to a remote computer. REMOTE gives you access to the new, inexpensive time-sharing systems designed for home computer users, such as MICRONET or THE SOURCE, or any other time-sharing system. You can also use REMOTE to talk to another microcomputer system. All you need is your FLEX based 6800 or 6809 disk system, a serial interface, a modem, and REMOTE.
REMOTE is flexible. You can control its many features and options thru local commands at any time. Output from the host system (from the phone line) may be directed to the console CRT, the system printer, or a disk file, or any combination of the three. You can change parity checking, control character definitions, etc. You can read files from disk to the phone line under user control. You also can issue any FLEX command while you are not receiving data.

REMOTE is a terminal program designed to give you the utmost in flexibility and convenience with as many different timesharing systems as possible. It was written by Tom Speer, P.E., an Electrical Engineer with over twelve years of data communications hardware and software experience.
Remote will support the New Thomas Instrements Modem Card.

INTRO PRICE $39.95

NEW

READTEST
English Text Analysis Program
By Dale Puckett

READTEST is a must for all writers and writing instructors. Reads prose from disk file and tells how well it was written. Reports number of lines, words, sentences, personal words, affixes, average sentence length. Individual reports pinpoint trouble areas. Overall index tells who can read it and who would print it. Fast 68XX object code. Runs in FLEX™. READTEST, a 68XX assembly language program written by 68 Micro Journal Contributing Editor Dale L. Puckett, is such an indispensable applications program for all writers, fiction or non-fiction, student or professional. READTEST reads your letters, short stories, news stories, reviews, novels, etc., from the disk file you prepared them on and tells you how well they were written. READTEST is a powerful tool designed to aid both the student writer and keep a running check on the experienced writer. It is a serious application of your 68XX microcomputer that will pay for itself with the first check you receive for a readable manuscript.

Approx. 12K: INTRO PRICE UNTIL JAN. 1, 1981
See Review in August '80 "68" Micro $39.95
REGULAR PRICE $54.95

ESTHER
An exercise in artificial intelligence
By Dale Puckett

ESTHER is Eliza plus. Artificial intelligence in pure 68XX code. Her source shows you how. Her object will amaze your friends. ESTHER: remembers names, drops them, uses the player's name, and even echos keywords. ESTHER identifies more than 75 keywords and uses almost fifty sets of replies. A few of the sets contain as many as 21 replies to help her avoid redundancy. ESTHER features auto line length and runs in FLEX™. She obeys TTYSET. She is both educational and fun. ESTHER, written by 68 Micro Journal Contributing Editor, Dale L. Puckett, is the result of a two year long experiment with artificial intelligence in 68XX assembly language programming. ESTHER randomly inserts the players name in the conversation. Occasionally, she uses part of the player's reply in the middle of her answer or next question. ESTHER has the ability to echo keywords. This allows her to respond to replies from the player which are in the third person.

ESTHER identifies proper nouns and uses them in her replies. She also saves them for later use.

APPROX. 16K : INTRO PRICE $39.95

FLEX is a trademark of Technical Systems Consultants Inc

master charge

VISA

FRANK HOGG

DENTAL LABORATORY
130 Midtown Plaza
700 East Water St.
Syracuse, N.Y. 13210
(315) 474-7856

All software is currently available on Flex™ 2.0 5" soft sectored disks and DMAF 8" Flex disks. The package includes: a users manual, disk with object code, FULLY COMMENTED SOURCE LISTING, a programming manual with information about the program, hints for changes and where applicable, example programs.

VISA and MC accepted. SOURCE TCF339
DEALER INQUIRY ENCOURAGED. Contact Frank Hogg for more information.

ATTENTION PROGRAMMERS!
We are looking for quality software to market. Contact Frank Hogg.

** JCP is available in 6800. Please call regarding 6809 availability.

Flex User Notes

UNFOLDING A DISK

Have you ever received a disk nicely folded or crushed by the U.S. Post Office? I have received several in that condition, and recently I sent one to a reader who received it in non-working condition, and wrote asking me if I had any techniques for recovering such a disk. I have had to perform surgery on a disk several times, and have always been successful in recovering the information. First of all, don't throw away the next disk that you find worn out or badly scratched (the magnetic medium) but save the outside jacket. First, separate the "flap" of the jacket at the end opposite the slot for the read/write head access. You will find this flap is "welded" or heat sealed together. After opening this end of the jacket, carefully cut the flap off entirely and remove and throw away the magnetic disk inside, that was bad to begin with.

When you receive a disk that has the jacket creased or crushed so the disk won't turn inside, do the same thing to it, being very careful not to damage the magnetic disk inside. Then, remove the disk from the jacket and carefully place it in the previously prepared good jacket. Be careful to place it in the jacket in the same orientation in which it was removed from the damaged one, ie. don't "flip" it. Now, if the disk itself has not been damaged, you will be able to copy it. In general, it works best to close the door of a drive (which engages the drive with the center of the disk) with the drive motor running. This helps center the disk on the spindle. It also prevents the "crinkling" around the edge of the center hole of the disk. Since the disk in this case is free to back out of the jacket, it is more important to be sure the drive is running when the door is closed. I've used this technique to rescue among other disks, my first copy of FLEX2 which SWTPC shipped inside the instruction manual with no cardboard stiffeners. The mailman folded the 9 by 12 envelope in half and stuffed it in the mailbox. I also recovered my first copy of Lucidata Pascal this way, and I have used the technique a couple of times on disks sent by readers.

If you do this, I strongly recommend copying the information onto a new disk that is in good shape. A disk running in an open jacket for any length of time will accumulate dust and dirt and wear out or become damaged much sooner than one in a closed jacket.

MORE ON PASCAL

Due to favorable response on the short item on Pascal and ways of checking for variables being within reasonable range, I offer this time, a program called DATE. DATE does not contain all the features of Pascal, and indeed, it would be hard to write a useful example program that does. However, DATE is a relatively simple program that illustrates well the use of several features of the language. A Pascal program always takes the same form or outline. The first line must be PROGRAM followed by the name of the program. The following sections must be in order but any section may not be present in a given program:

```
PROGRAM (PROGRAM NAME);

LABEL 10,20,30;

CONST

  PI = 3.14159265;
  NUMBER = 3;

TYPE

  DAYS = 1..32;

VAR
  CH : CHAR;
  N  : INTEGER;

PROCEDURE PROCEDURENAME (PARAMETER : INTEGER);

  BEGIN
    (* PROCEDURE IS A BLOCK WITH ANY OR ALL OF
      THE ABOVE DECLARATIONS. ANY VARIABLES
      DECLARED HERE ARE "LOCAL" TO THIS
      PROCEDURE. *)

  END;

(* THIS IS A COMMENT. THERE MAY BE ANY NUMBER
  OF PROCEDURES AND A PROCEDURE MAY "CALL"
  ANOTHER PROCEDURE. ALL CALLS
  MUST BE TO PREVIOUSLY DECLARED PROCEDURES *)

(* AFTER ALL PROCEDURES, THE MAIN PROGRAM STARTS *)

BEGIN

  (* BODY OF MAIN PROGRAM HERE *)

END.
```

Note that in the program DATE, there are no constants declared nor are any labels used. A label implies the use of a GOTO somewhere in the program. Most of the proponents of structured programming feel the GOTO to be a no-no. There are a few instances where avoiding a GOTO is so complicated that it makes the program less structured than using one. For these cases, the GOTO has been provided. It should not be abused. DATE also doesn't have any constants declared. It does, however have some TYPE declarations. MONTHS is declared as an "enumerated" type having 12 possible values, all of which are listed in the declaration. Pascal not only "notes" these 12 values but their order may be used later in the program. The variables DAYS and YEARS, are called subrange types, in that they are really both of the type INTEGER, but they may not be assigned or otherwise take on the whole range of possible integer values. DAYS is limited to the values from 1 to 32, and years from 0 to 99. The VAR declarations define the variables to be used in the program. MONTH is of the type MONTHS, DAY of the type DAYS, and YEAR of the type YEARS. There are some predeclared types in Pascal, namely INTEGER, REAL, CHAR, and BOOLEAN. Of course an integer is what we call a whole number, and a REAL contains a decimal fraction or an integer plus a decimal fraction. Constants take on the type of the value assigned to them. Thus in the example above, PI is a REAL variable and N is an INTEGER. If we declared a variable SWITCH = TRUE; we would have defined SWITCH as a BOOLEAN constant. BOOLEAN variables have the value TRUE or FALSE. We could declare our own type BOOL = (ON, OFF); and use these values rather than TRUE and FALSE.

This brings us to the first procedure in the DATE program, PROCEDURE ENTERDATE; This procedure has no parameters passed to it. It operates on the variables that have been declared as part of the main program, DAY, MONTH, AND YEAR. It prompts for the month number and inputs a response from the terminal. It then enters a loop, after assigning MONTH the value JAN. The loop increments a counter K until it is equal to N, the month number, and "increments" the variable MONTH through the list of months with the instruction MONTH := SUCC (MONTH);. The SUCC means SUCCESSOR or next value in the list of values assigned in the TYPE declaration. If we were to use SUCC (DEC); we would be in trouble since DEC is the last value and the list is not

assumed to "wrap around". In any case, we have now assigned the value of the month to MONTH. The WHILE DO loop tests for the condition at the start of the loop, which means it may not be executed at all (if n=1). Before this loop, K is set to 1. If the Month input was JAN, we don't want to get into the increment loop, and the WHILE DO takes care of that. The ENTERDATE procedure similarly and in a simpler manner gets the day and year from the operator. WRITELN; all by itself, is the same as PRINT by itself in BASIC. It simply causes a linefeed and carriage return.

The procedure INCREMENT is more interesting. It "figures out" whether DAY has been incremented past the end of the current month, and in that case, updates the month and day. We have not here updated the year on DEC 32, though that would be very simple. In fact, one could test for the year being divisible by 4 and allow for FEB 29 in a leap year. To continue with the program, DAY is first incremented. The program then tests for the shortest month, a single case, with an and of the conditions MONTH = FEB and DAY = 29. If this test is true, the DAY is set to 1 and the MONTH to MAR. The next test is for the 30 day months, and its working should be obvious by now. If the DAY gets to 32, obviously the month is not one of the shorter ones, and this test is sufficient to reset the DAY to 1 and increment the month. Note that DEC is trapped by the IF MONTH = DEC, and the other months are taken care of by the ELSE MONTH := SUCC(MONTH);

Although an enumerated type like MONTH may take on the values listed, the simple command WRITE (MONTH); doesn't cause the printing of the value of MONTH to the terminal. Instead, the procedure SHOWDATE must be used. The Pascal CASE statement is the same as the ON N GOTO statement in BASIC, except that you don't have to calculate a value 1,2,3,etc. for the "on" variable. The structure is adequately illustrated in the program. The syntax is CASE VARIABLENAME OF, followed by a list of possible cases as labels, with the consequence following each label. The result is like a series of IF-THEN statements, IF MONTH = JAN THEN WRITE ("JAN"); IF MONTH = FEB THEN WRITE ("FEB"); etc. Standard Pascal gives an error message if in this case for example, MONTH has a value other than one of those listed in the case statement. In this program, we have used all the possible values of MONTH so that is not a possibility. Some of the Pascal implementations have attempted to "fix" this by adding an OTHERWISE to the end of the case structure to catch any other values not listed in the case structure.

The last line of SHOWDATE is the write statement. WRITELN followed by a list of variables of literal strings in quotes, is like a PRINT in BASIC without a comma or semi-colon at the end. That is, a linefeed and CR are generated at the end of the line. In this case, we have already output the MONTH, and this line adds the value of DAY and YEAR. The ":2" following the variable name DAY and YEAR specifies tha the integer number is to be printed in a field of 2 columns.

In the case of his example, the main program is really nothing but a test routine for the procedures. If these procedures were to be included n a larger program, the main program there would probably drive these procedures perhaps asking for the date on power up, and updating the date on the basis of a real time clock reaching Midnight. The REPEAT UNTIL structure looks for the condition after the UNTIL to test TRUE. The condition in the main program here is essentially REPEAT UNTIL FALSE = TRUE, which of course can't happen. This is therefore an "infinite" loop that may be exited only by a reset or power off condition.

I hope this quick tour through a Pascal program will point out the fact that the main structures of Pascal will be familiar to you, and the great flexibility in defining data types will make the program readable to you and to another programmer. The line in the main program IF CH = "E" THEN ENTERDATE ELSE INCREMENT; calls either the procedure ENTERDATE or the procedure INCREMENT depending on the value of the character entered from the terminal. A procedure is called simply by naming it.

You may be wondering about the use of the semi-colon. It is at first confusing, but it simply ends a statement. In a few cases, the statement is terminated by another means and the semi-colon may be omitted. A compound statement is one that starts with a BEGIN, containing two or more simple statements, that themselves end with a semi-colon, the whole compound statement ending with an END;. The END terminates the compound statement and the statement preceeding it does not need the semi-colon, but most Pascal implementations will not object if one is there. The last statement before END in a CASE statement must not have a semi-colon, and you must be careful not to put one in the middle of a statement such as FOR N:= 1 TO 10 DO; WRITE (N:2);. With the semi-colon after the DO, the loop will simply "count to itself" to 10 and then execute the WRITE once. Without the extra punctuation, the whole thing becomes one statement, and the loop terminates at the proper place. I hope this little run through a Pascal program will convince you that it is not so formidable after all.

A REQUEST

Some of you may have noticed that I haven't included my phone number in the heading for this column. My number is listed in Ann Arbor, and the information operator will give it to you if you ask for it. when I was publishing a newsletter for 60 or so people, I received calls infrequently, and there was no problem. '68' Micro Journal has a circulation of about 10,000. If one per-cent of you were to call me in one week of evenings, I would be driven slightly buggy. Please remember that this activity is in addition to a full time job, and respect my "I want to be alone" time in the evening. I won't turn down a long distance call, though I may be irritated by it. If you have a "real emergency" please feel free to call. Otherwise, please allow me my time with my family and permit me the option of answering your question or helping with your problem at a time that is convenient to me. I have so far answered every letter I have received from you readers. I don't know if that will always be possible, but I will continue as long as I can. Please write if you have questions, problems, comments, criticisms, etc. Some of the best material for this effort has resulted from your input of questions. If you have a particularly perplexing problem, perhaps many others are experiencing the same problem and would like to have the answer too, or maybe one of our readers has solved the same problem for himself, and a mention of it here will bring a very good solution to you.

A CLARIFICATION

In the September issue, there is an article by Wilton Hart that provides a very good patch to FLEX2, which I have implemented in my system in a slightly different manner than that described in the article. My doing this was motivated by a letter from John Deal, one of our readers. John had first tried appending the patch to FLEX.COR, and for some reason had no success. He then tried loading FLEX2, adding the patch and saving FLEX2. Probably because FLEX2 had already been through the initializing portion, asking for the date and handling the STARTUP file, his saved version didn't ask for the date. Wilton indicated that the boot program only loads FLEX2 and will not load the patch if it is placed at $BFBI as he has done. The boot program, I assume only loads to $BFB0. Rather then perform a permanent modification to FLEX2, I chose to prepare the patch which I called FLEXOV, as a separate file with no transfer address. I added my previously done overlay to change the head seek rate for my Shugart SA-400

drives, and modified the STARTUP file by adding 'GET FLEXOV.BIN.0' as one of the instructions. This of course loads the file, and it doesn't matter if the load overlays FLEX2, since now the Boot program is long gone.

There is another advantage to this approach, in that FLEX2 is not modified permanently. Your FLEX2.SYS file is left intact, and you will not have "different" versions on different disks, and additionally, you won't have the problem of having to figure out how to "unappend" FLEX2 if you ever want to undo the patch. To remove it all you need do is delete the instruction in the STARTUP file to get it. Another advantage is that you may combine your other patches with this one and save sectors on your system disk. If yours is like mine, you have very few sectors left on it. This same approach, of course applies to Miniflex as well. The listing of my overlay is included here. By the way, thank you Wilton Hart for a super fix to Flex2!

```
PROGRAM DATE;

TYPE
   MONTHS = (JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC);
   DAYS = 1..32;
   YEARS = 0..99;

VAR
   MONTH : MONTHS;
   DAY : DAYS;
   YEAR : YEARS;
   CH : CHAR;

PROCEDURE ENTERDATE;

VAR
   K,N : INTEGER;

BEGIN
   WRITE ("ENTER MONTH (1..12)");
   READ (N);
   WRITELN;

   MONTH := JAN;  K := 1;
   WHILE K<>N DO
   BEGIN
      K:=K+1;
      MONTH := SUCC (MONTH)
   END;

   WRITE ("ENTER DAY (1..31)");
   READ (DAY);
   WRITELN;

   WRITE ("ENTER YEAR (0..99)");
   READ (YEAR);
   WRITELN;
END;

PROCEDURE INCREMENT;

BEGIN

   DAY := DAY+1;
   IF (MONTH = FEB) AND (DAY = 29) THEN
   BEGIN
      DAY := 1;
      MONTH := MAR;
   END;

   IF ((MONTH=APR) OR (MONTH=JUN) OR (MONTH=SEP)
      AND (DAY=31) THEN               OR (MONTH=NOV))
   BEGIN
      DAY := 1;
      MONTH := SUCC (MONTH);
   END;

   IF DAY = 32 THEN
   BEGIN
      DAY := 1;
      IF MONTH = DEC THEN MONTH := JAN
      ELSE MONTH := SUCC (MONTH);
   END;
END;

PROCEDURE SHOWDATE;

BEGIN
   CASE MONTH OF
      JAN : WRITE ("JAN");
      FEB : WRITE ("FEB");
      MAR : WRITE ("MAR");
      APR : WRITE ("APR");
      MAY : WRITE ("MAY");
      JUN : WRITE ("JUN");
      JUL : WRITE ("JUL");
      AUG : WRITE ("AUG");
      SEP : WRITE ("SEP");
      OCT : WRITE ("OCT");
      NOV : WRITE ("NOV");
      DEC : WRITE ("DEC")
   END; (* CASE *)

   WRITELN (" ",DAY:2,", ",YEAR:2);
END;

(* MAIN PROGRAM *)

BEGIN
   REPEAT
      WRITE ("ENTER DATE(E) OR INCREMENT(I)?");
      READ (CH);
      WRITELN;
      WRITELN;
      IF CH = "E" THEN ENTERDATE ELSE INCREMENT;
      SHOWDATE;
   UNTIL FALSE;
END.
```

RONALD W. ANDERSON
3540 STRUBRIDGE COURT
ANN ARBOR, MI 48105

Left to right: Dan Farnsworth of Palm Beach Computers presents a new GIMIX mainframe to Grand Prize winner Brian F. Bailey of Plantation, Flordia.

Brian Bailey of Plantation, Flordia was presented with a new GIMIX Standard S50 Bus computer (value $900.00) at the November meeting of the South Florida Computer Group. This fine machine was donated by GIMIX, 1337 West 37th Place, Chicago, IL 60609.

Brian was a recent Grand Prize winner in the International Giant Software Contest, sponsored by 68 Micro Journal and various manufacturers and vendors of Standard S50 Bus computers and software.

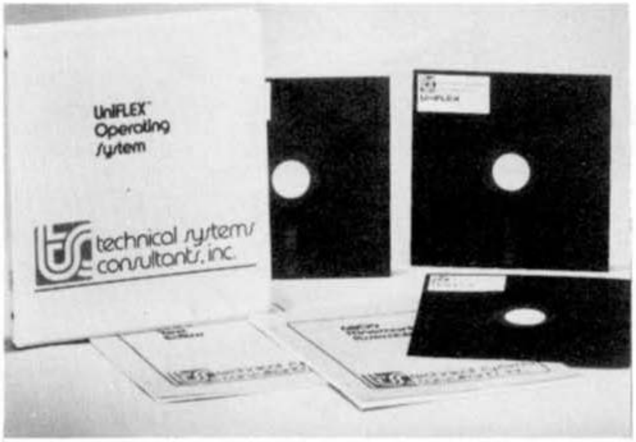
Bailey's winning program is to be made available by GIMIX for their customers. It will also be published in 68 Micro Journal at some later date.

Bailey is an employee of Circle Line of New Yrok as Chief Marine Engineer. Bailey formily attended Florida

Technical University Orlando and RCA Institute in New York. His hobbies include computers, ham radio and music.

Our congratulations to you Brian for a fine software utility.

UNIFLEX



IV. The File System

The UniFLEX™ operating system has three main functions, file maintenance, I/O control, and task scheduling. The structure of the file system is probably the most important, since design flaws here will impair almost every program run on the system. Here again, the UNIX™ file system was modeled quite closely.

There are three basic types of files, ordinary, directory, and special. The majority of files are ordinary files. These files are simply a collection of bytes, having no special meaning. There is no concept of 'records' and no forced structuring of data. All files may be accessed either sequentially or randomly and may be as large as one billion bytes.

All files in the system are 'protected' by a set of permission bits. These permission bits determine whether or not a file may be read, written, or executed. Two bits exist for each of these modes, one defining the permission for the file's owner, and another one for the permission of all others. As an example, the owner of a file may set the permissions such that he may read or write the file, but all others may only read it.

The second file type is the directory. A directory is exactly the same as an ordinary file with the exception that the data in the directory is operating system defined. Each directory entry requires 16 bytes, 14 of which are used to store the file name, and the remaining two are used for the 'File Descriptor Node', or fdn for short. The fdn is simply a 16 bit number used to identify the file on the disk. There is no limit to the number of directories.

The directories on the system form a hierarchical tree structure. The root of the tree is called the 'root' directory. Any directory may contain entries which are names of other directories (or subdirectories). Each user of the system is assigned his own directory. When a user 'logs-in', this becomes his 'current directory'. Since many files and directories exist on the system, a mechanism is needed for specifying a particular file in a specific directory. This mechanism is known as a 'path name'. The path name is a list of directory names separated by slashes, all followed by the file name desired. As an example, the path name '/usr/john/test' tells the system to start in the root directory (specified by the leading '/' in the path name), find the directory named 'usr' in the root, then scan that directory for the directory named 'john', and finally scan the directory 'john' for the file named 'test'. If a path name is specified without the leading '/', the search will start in the current directory as opposed to the root directory.

All directories have at least two entries, one named '.', and one named '..'. These names are purely convention. The file '.' represents the directory itself, and the file name '..' represents this directory's parent directory. The '.' entry is useful in referencing the current directory without knowing its name, and the '..' entry is used for reverse traversal of the directory tree.

The permission bits previously described also apply to directories. If a user 'read' protects his directory, others will not be able to display the contents of the directory, and if the directory is 'write' protected, no new files may be placed in the directory. If a directory is 'execute' protected, it may not be 'searched' for a specified file name, or as part of a path name.

As an extension to the directory tree structure of a file system, another file system (disk unit or units) may be 'mounted' at any node of the tree. The mounting process effectively replaces an existing node (directory) with the root directory of the mounted file system. As an example, a system with two disk drives will use one of the drives as the system 'root device', that is, the drive containing the directory known as '/' to the system. In order to access the directories and files on the second drive, it is only necessary to mount this device on an existing directory of the root device. The mounting operation will cause the contents of the selected directory to become inaccessible, replacing its contents with the root of the directory tree on the second drive. An 'unmount' operation will restore the original directory. This procedure logically extends the notion of file names to allow access to any file on any currently mounted file system.

A specific example will clarify the mount operation. Let's assume there is a directory named 'user2' in the root directory of the main system disk. Let's also assume that we have another disk which contains a file named 'test' in a directory named 'source' in the root directory of that disk. Performing a mount of this second disk onto the directory 'user2' will now allow access of the file 'test' with the following path name:

/user2/source/test

Note that no mention of 'device name' or device type was necessary to access this file. This structure allows several file systems to be connected together as one big tree, greatly simplifying overall file organization.

The third type of file in UniFLEX™ is the device (or special) file. All devices on the system appear as file names in directories, just as regular files. All of the devices are normally kept in the directory '/dev'. This means that programs which read and write file data may just as easily read and write data to and from a device. As an example, to write data to a printer, the program could write to the file '/dev/printer'. Treating I/O devices in this way allows fairly device independent I/O, in that file and device I/O operations are very similar. It also allows the same protection scheme used for files to work for devices. This mechanism of device

files, or 'special files' is identical to that used by the UNIX™ operating system.

Since files and I/O devices are so similar, the same system I/O calls may be used for both. The UniFLEX™ system calls to perform I/O allow 'files' to be created, opened, read, written, and deleted. The following examples show the calls as procedure calls in a general high level language form. The exact calling sequence is defined by the actual language in use. The call to open a file looks like this:

```
open(name, mode)
```

where 'name' is the path name of the file to be opened and 'mode' specifies whether the file should be opened for read, write, or update (both read and write). The open call returns a value called the 'file descriptor' which is used to identify the file for future I/O operations. The file descriptor is simply a number which the operating system associates with the file opened.

The open call requires the specified file to already exist. To create a new file (or truncate an existing file to zero length), the 'create' system call is used.

```
create(name, permissions)
```

This call also returns a file descriptor. The 'permissions' argument specifies which permission bits should be associated with the file. Once the create has been executed, the file is left 'opened for write'.

To read data from an open file, the system call 'read' is used.

```
read(file-desc, buffer, count)
```

The 'file-desc' is the file descriptor returned by the open call. The argument 'buffer' is a pointer to the space where the system will place the data from the file. The 'count' argument specifies the number of bytes wanted from the file. The corresponding 'write' operation is similar.

```
write(file-desc, buffer, count)
```

In this case, 'count' bytes are written from 'buffer' to the file represented by the file descriptor. In both the read and the write calls, a value is returned which is the actual number of bytes read or written. When writing, the returned value should always be equal to the requested 'count', or an error has occured. The value returned by read does not need to equal the 'count', and a returned value of zero represents the 'end of file' condition.

Reading and writing may take place in any part of the file. Each open file has a 'file pointer' associated with it. Reads and writes start at the current position of this pointer and advance the pointer by the number of bytes transferred. An open operation sets the file pointer to the beginning of the file. The 'seek' system call allows repositioning of the file pointer. It has the form:

```
seek(file-desc, offset, type)
```

where the file descriptor selects the file, and the 'offset' is a byte count representing the relative position from the file's beginning, end, or current position, determined by the value of 'type'. This call returns the actual value of the resulting file pointer (bytes from the file beginning). Seeking beyond the end of a file and reading will result in an end of file condition, while writing will simply extend the file to include the written data. It should be noted that file extensions allocate just enough disk space to record the new data. As an example, performing a seek to byte 10,000 in a file which has length of 100, and writing one character will produce a file of logical length 10,000, but only two disk blocks will be allocated to the file. Reading data from the file will yield null bytes where no disk space is actually present.

The disk I/O facilities of UniFLEX™ are quite efficient, allowing full processor overlap with disk I/O transfers. The system maintains a disk block buffer cache used to keep the most recently accessed disk blocks in main memory. When a program requests data from a particular disk block, the system first searches its memory buffer cache for the block. If it is found, no disk transfer need be made. If it is not found, the oldest block in the cache is given up, and its corresponding buffer is replaced by the contents of the requested block.

UniFLEX™ also supports full 'read ahead' and 'write behind' data transfers. Read ahead implies that whenever the system needs to read a block of a file, it will automatically read the next sequential block as well. Since the disk read operation is overlapped with the CPU operation, very little, if any, time is wasted doing the additional read. Write behind means that any data to be written to the disk is simply placed in one of the cache buffers, and written at a convenient time. Programs writing data are not delayed until the write actually occurs. This combination of read ahead, write behind, and the block buffer cache, gives UniFLEX™ a superior I/O transfer rate.

UniFLEX™ also supplies a mechanism for file 'record locking'. This is one area where the UNIX™ operating system falls short. The system call:

```
lrec(file-desc, count)
```

will lock 'count' bytes from the current file pointer in the file represented by the file descriptor. The count size or record size may be anywhere from 1 to 65535 bytes. The locking action is more of a convention than an actual hard lock operation. After locking a section of a file, other programs may still read or write that section of the file without error. If another program tries to lock a section of a file which is already locked, however, an error will result. This structure has proven to be very efficient in that programs dealing with data base type files may make use of the lock mechanism and preserve data integrity, while those working with regular files need not be concerned. A locked record may be unlocked by another lock call, closing the file, or issuing the 'urec' call to specifically unlock the record.

There are several additional system calls in UniFLEX™ pertaining to I/O. These include file closing, deletion, and linking. Other calls exist to create new directories, change a file's owner and permissions, and get a file's status.

V. Task Structure

Each program under UniFLEX™ runs as a separate task. When a task is actively running, it has its own dedicated address space. This means that the task has the complete address space of the CPU and any part of this space will either contain memory or be

totally void. No I/O devices or system code is present when the task is running. Each task is assigned enough memory to hold its program, data, and stack. The program (or text) size is set at the initial excution of the task and remains fixed. The data and stack segments may grow or shrink dynamically. The text part of a program may be 'shared' among all tasks currently executing the same program. This is done automatically and tends to make more efficient use of available main memory. The operating system keeps a large amount of information about each active task, including which user started the task, the task identifier, the current program size, amount of CPU time used, age of the task, and task activity information. Tasks are scheduled CPU time based on their priority. The priority value is constantly adjusted by the system to reflect the current status.

New tasks are created by the 'fork' system call. The fork call causes the calling task to duplicate itself, or split into two identical tasks. The complete address space of the calling task is duplicated for the new task, as well as the task's complete environment, including open files, etc. The new task starts execution upon return from the fork call. It may be distinguished from the parent in only one way. The fork call will return a value of zero to the new child task, and a value which represents the child's task identifier (never zero) to the parent. This allows each task to determine if it is the child or the parent. The return from the 'fork' is a little different at the assembly language level. Here, the return to the original task is two bytes beyond that of the new task. This allows the new task to perform a 'branch' instruction before continuing. The child's task identifier is still returned to the parent task.

There are no restrictions placed on what the new task can do. Normally, it will perform an 'exec' system call which will invoke a new program. The form of the 'exec' call is as follows:

```
exec(file-name,argument1, argument2,..., argumentn)
```

The 'file-name' is the name of the program to be loaded and run. The calling task's address space is replaced by that of the called program. The 'arguments' are made available to the new program as an array of strings. Note that a return from an 'exec' to the calling task is an error condition, usually because the specified file name was not found or not executable. The 'exec' call can be thought of as a 'jump' type instruction where control is passed to the first instruction of the called program. Most of the task's environment parameters, such as open files, are preserved across the exec. Leaving files open allows for easy implementation of the standard I/O mechanism. All tasks usually start with three files already open known as the standard I/O files, as previously described. These files have file descriptors 0 for the standard input, 1 for the standard output, and 2 for the standard error channel.

A task which 'forks' another task may 'wait' for the child task to terminate. The wait system call will block the calling task until one of its children tasks terminate. Upon termination, the wait call will return to the caller, returning the task identifier and the termination status of the dead task. Tasks normally terminate by the 'term' system call. It has the form:

```
term(status)
```

where status is a value made available to the parent task. A status of zero indicates normal termination, while nonzero specifies an error condition. A task may also be terminated by a 'program interrupt'. Tasks have a choice of ignoring or catching these interrupts to avoid termination. As an example, the interrupt character (control C) is sent as a program interrupt to all tasks associated with the terminal producing it. Normally, this will terminate the task, but programs like the Text Editor choose to catch this interrupt and take special action such as re-issuing the prompt to accept another command.

Tasks are run on a prioritized basis, the highest priority always being run. A task's priority is constantly being adjusted to reflect its size, age, and CPU activity. Tasks may also be swapped to secondary storage if the demand arises. The swap algorithm has built in hysteresis to avoid swapping out a task which has just been swapped in but not permitted to run. UniFLEX's™ scheduling routine is quite complex and tries to take in as many factors as possible when making scheduling decisions. As an example, tasks which have been ignored for a long time tend to increase in priority, and those which are hogging the system's resources are penalized. The idea here is to be as fair as possible to all tasks in the system. There is only one system imposed limit to the maximum number of tasks permitted in the system at any one time, the amount of memory available for the 'task table'. This does not tend to be a restriction since other hardware limitations tend to determine the useful maximum.

There are several other system calls which pertain to tasks. These include calls to get a task's identifier, its owner, and one to incrementally adjust the priority over a small range. This last call is particularly useful for setting lower priorities for tasks which are typically background jobs.

VI. UniFLEX™ Overview

UniFLEX™ is a very complete multi-user, multi-tasking operating system. It is intended to run with larger microcomputer systems and is not well suited for the small memory, small disk systems. The decision to require memory segment management (not bank switching) and efficient disk devices eliminated all compromises in the design. Small machines should have small operating systems while sophisticated hardware configurations deserve nothing but the most sophisticated operating systems. Trying to write an operating system which works equally well with limited hardware configurations almost always results in a less than optimal system.

One question which always arises when discussing multi-user operating systems is 'How many users'? This is a difficult question to answer because there are so many variables. UniFLEX™ can support any number of users, but the practical number ranges from two to about twenty on the 6809, and up to thirty-two on a 16 bit microprocessor. In most environments, more terminals may be connected than the upper practical user limit since not all terminals will be in use at any given time.

Many factors determine the maximum user count. These include such things as the amount of main memory, the processor clock speed, number of different hard disk drives, number of hard disk controllers, the hardware I/O structure, the efficiency of the memory management unit, and response times desired. The amount of main memory affects the amount of swapping the system will perform. If a separate high speed disk is used for swapping, less main memory is required. If one disk is being used for all system and user files, as well as swapping, additional memory will speed up the system significantly. The speed of the swapping disk is also very important. Those running with a floppy disk drive for swapping will see a definite decrease in system performance.

Some applications are very terminal I/O bound. Word processing is one example. A system will generally be able to support more terminals for word processing than the same system could support for scientific or engineering applications. Business applications also tend to be very terminal I/O intensive. Keep in mind, that a terminal which is running a program waiting for input, has almost no impact on the system. Those environments which present this condition the majority of the time will be able to support many more users than those which are constantly running compute bound programs. Programs which generate a tremendous amount of output will degrade the system if the output is displayed at high baud rates. This degradation may be overcome by a 'front-end' I/O processor.

The final consideration in determining the number of users is the response time required. Response time is defined as the interval of time from the instance a keyboard entry is made, until the expected response is obtained. In many environments, the response time is not critical. Many educational systems, for example, would rather support more users at the cost of response time, since more users reduces the cost per student. All of these considerations are not peculiar to UniFLEX™, but apply to any multi-user system, regardless of size.

The efficiency of an operating system can be partly determined by the amount of overhead required to perform a particular operation. UniFLEX™ was designed to keep system overhead at a minimum. Much of the current overhead is hardware imposed, but future systems promise to improve on this.

Since file activity is usually the biggest bottleneck in multi-user systems, the file system must be very efficient. UniFLEX™ is very efficient, not only in file storage overhead, but also in file transfers. The overhead involved in file storage is determined by the directory space, the file status information, and the file mapping information. In all, this is typically less than 8% overhead, a figure which is very respectable.

The disk transfer rate is where UniFLEX™ really shines. As a comparison, consider the test presented in 'The Bell System Technical Journal', July-August 1978, pages 1950-1951. This test compared three mini-computer operating systems by simply timing a disk file copy. The file was 480 blocks in length (245,760 bytes) and was copied on a system which was otherwise idle. This same test was run under UniFLEX™, on a Southwest Technical Products S/09 6809 computer system. The main system disk was a Century Data Marksman, which is Winchester technology and holds approximately 17 megabytes of formatted data. The 6809 was only running at one megahertz. The results of the test were as follows:

system	seconds	msec./block
UniFLEX™	27	28.1
UNIX™	21	21.8
IAS™	19	19.8

Both UNIX™ and IAS™ were running on DEC PDP 11/70's. It is no surprise that UniFLEX™ places last, but it is a surprise that it is only about 23% slower than UNIX™ on an 11/70! Increasing the speed of the processor to two megahertz should bring this value even closer (the total time would probably be reduced to about 24 seconds). This test does not prove much, if anything, but it is an interesting comparison.

This document is not intended to be a complete description of the UniFLEX™ operating system. Instead, it presents some of the system's highlights hopefully of interest to the reader.

RUMORS

New from SWTPC are two new CRT terminals, see inside front cover this issue. We have been using a couple for the past month or so and the sturdy construction and compatibility with the older CT-82 make either a simple and logical upgrade. There are some very nice improvements in utility as opposed to the CT-82, more on this in a review to come later.

The MICRO WORKS has developed and will soon be delivering some new machine language software for the TRS-80C™.

The primary program, delivered on tape, is a machine language monitor called CBUG.

Also available soon but not tested as of this writing is a disassembler for 6809 code, which allows disassembly of any program in RAM, including BASIC, or any program pak. Program paks may be disassembled by covering a pak pin with tape and having the pak plugged in, more on this maybe next month as we want to test this to insure that no damage will occur.

My understanding is that the first offerings of the 'Dissambler' require a printer (serial) attached to the output port for listing. This is due to the required width of a listing and the restriction (mandated by the 6847 generator) of 32 characters screen width. Bob, of THE MICRO WORKS informed me that a CRT screen version is soon to follow.

Also will be a general purpose pak board for insertion of your own (or purchased) programs in EPROM, 2716. This inserts in the pak-slot on the side.

Information has it that the present version of the SAM 6883 IC is experiencing a larger than expected failure rate. If you need to order another they are already in the Tandy spare parts catalog.

As of this writing we know of TRS80C™ machines that are expanded fully with a combination of dynamic and static RAM, 40K of useable memory. Also 'uploading and downloading' from the color computer to your Standard S50 bus machine is possible using CBUG.

The level II of BASIC is not available as of this writing but expands the graphics by commands such as DRAWLINE, PAINT, STRING and includes the trig and additional string functions, among other extended commands and functions.

These programs allow machine and assembler language programming and also preserve calls to the TRS80-C™ BASIC.

I have found no serious bugs in CBUG and it is a powerful monitor, as the listing above indicates, especially for it's small size.

A real flurry of advertisers for color computer software has come to us within the past few days. Some we are not accepting until we 'check it out', as is our standard policy.

Even Mickey Ferguson, well known 68XX author and occasional contributing editor to 68 Micro Journal, is busy preparing color graphics software for the Tandy color machine. This has led to the formation of 'COLORWARE™', devoted to serious and fun software for the TRS80C™. Good luck Mickey!

Hope to have a review of the Computerware offerings next month or so. See advertisement this issue.

If you are planning to develope and advertise software or hardware for the TRS80C™, then you should drop me a line. We have received a healthy batch of new subscribers, who have obtained a color computer. They will need to know.

BOOKS

Over the past few months we have been receiving books for review. This month we will look at one, from TAB, that many users will find useful. It is a softback titled, THE MOST POPULAR SUBROUTINES IN BASIC. It is TAB book number 1050, and is the effort of Ken Tracton.

This entire offering is a collection of those subroutines that require a specific knowledge to write. As the cover suggest it is 'not a theoratical manual, but a practical handbook for the professional and hobbyist'.

For those who have a problem with math at all levels it is a necessary programming tool. For the advanced programmer, not wishing to reinvent the wheel, it is also a necessary programming tool. Physics, chemistry, math, calculas, metric conversions, electrical and mechanical engineering, finance and business subroutines are only a part of this book. An awful lot of material has been packed into it's 183 pages.

It is this reviewers opinion that this manual at $5.95 is an excellent addition to any programmers library.

—

Index '1980

The following is an INDEX of articles and other material published in 68 Micro Journal, for the year 1980. Some back issues are available for $3.50 each plus mail and handling.

Wommack, page 8. MICROWARE OS9, by Kaplan, page 9. TSC UNIFLEX, by Shirk, page10. FLEX USER NOTES, by Anderson, page 13. WINNERS - Software Contest, page 15. SAVE THAT SWTPC MPA CPU, by Caudell, page 16. TRS80 TO S50 BUS, by Mayhugh, page 18. BIT BUCKET, page 22. F&D 5-8 DISK CONTROLLER, review, page 30. CLASSIFIED, page 31. HELP, page 32.

Please note that in the monthly column 'BIT BUCKET' appears, for the year, hundreds of small and not so small articles of special interest to most 68XX users. Most of these would comprise a full article in some magazines, but because they came to us as letters, etc., with listing we decided to place all the hints and kinks, fixes, suggestions on improving hardware and software and other valuable subjects in this one grouping. By appearing in BIT BUCKET in no way demeans an article or it's value. In fact, most all readers feel that some of the most valuable information published, appeared in BIT BUCKET.

radio shack COLOR

The first piece of software (canned) we have received for the TRS-80C™ is a ROM pak titled chess. The title flashed to the screen is that it is 'Microchess" version 2.0, by Peter Jennings. Also you are informed it is Copyrighted 1980 by Personal Software, Inc. On firing it up on one of our lab TRS80C™ I found that indeed it did display a chess board, in vivid color, with recognizable chess pieces. The graphics are good and the usa of color makes it interesting. So right into a game, I thought. However, at this point interest waned.

Nice graphics it's got, a good game, not so hot! Moves may be made from the keyboard or joysticks, that is if you have them. It has 8 levels of play. Supposedly from 1 to 8 with the difficulity factor going upward. Level 8 plays nearly as poorly as level one, it just takes more time doing it. In level one the skill level is about on a par with playing chess with a chimp, a slightly retarded one at that. The instruction book indicates that at level 8 it looks ahead three moves. In chessese that would signify 6 ply. A 6 ply chess game should should be capable of a game on the order of 1,000 to 1,400, rating (intermediate level). It don't! I really didn't expect it to play championship chess, but I did expect something better that what it has done so far. Some plays are apparently chosen by a 'random' move flag. Some time I felt that they all were. It seems to have a continuing 'death wish'. After a few games(?) I don't honestly know if I was more sorry for the poor game or just becoming more and more embarrassed by it's foolish antics.

It starts off fairly well (first couple moves) and from there on it gets worse and worse. No matter what the level, it seems to play some sort of 'point' game with no regard for board position (either side) or total points won by either side (it wins few from a player with any level of experience). Under some circumstances, haven't tried to figure what they all are, when it (or you) gets into what looks like trouble, it just gives up, erases the board and sets up for a new game. This seems to be the smartest move it has. Also it has no regard for a pawn on it's way to being promoted to a queen. It will if at all possible spend all it's time checking the king if possible, with no regard to other apparent useful moves. Despite all this, averone in the office plays it every chance they get. The fine graphics and joystick operations just make it plain fun. Also I must admit that I am biased when it comes to computer chess. * I know and expect good play, having a good version of Sargon 2.5 which does quite well considering it uses a 6502 (pardon me).

For a rank beginner it might be ok, for starters, but after a short while it leaves something to be desired. Needless to say I expect something better from a company like Tandy!

In all fairness it should be noted that this version (Microchess) was one of the first chess games to run on small computers. It's best point always was that it ran in a small amount of RAM. On the KIM™ it runs in less than 1K of RAM. There is no excuse for this version, as the TRS80C™ has more RAM available. Maybe it was a hurry-up project, these we have all seen before, from a lot of vendors. Maybe soon someone will offer a good game of chess for this machine. There certainly are quite a few floating around. * As stated in an earlier issue of 68 Micro Journal I mentioned as to how the 6809 is ideal as a chess playing CPU. It's excellent stack capabilities make it a natural. Strange some of the other chess game manufacturers have not wised up. If you are still with me, listed below are the commands per the book:

C - Choose or change colors.
L - Level choice.
J - Joystick control toggles on or off.
ENTER - Moves a piece.
X - Exchanges sides.
P - Force it to make next move.
T - Take back a move.
BREAK - Take back a move - also stops demo game.
A - To adjust a piece position.
E - Adds or changes a piece.
SHIFT C - Clears the board.
SHIFT R - Resign or restart.
SHIFT D - Run a demo game (it plays itself).
UP ARROW, DOWN ARROW, LEFT ARROW, RIGHT ARROW
Moves the position pointer.

Nowhere does the instruction book claim it plays a good game. It is hoped that the next offering along this line will!

Now for some good points. The TRS80C™ can be easily upgraded to 16K by the changing of two jumpers from the 4K to 16K positions, on the circuit board, and the swapping of the 4K RAM chips for 4116 dynamic RAM chips. Thats all there is to it! As stated earlier it is fun for the not too serious chess fiend.

The tape save and load speed of 1500 baud is an improvement. However, we found that the recorders we had used with other machines, would not work very well with the TRS80C™. So after purchasing another tape recorder, the one recommended by Tandy, we experienced little problem with the tape operation.

In the 'RUMORS' column I will try to keep you informed of what is becoming available for this machine. I receive a lot of mail, from all over the world, asking if we are going to support the TRS80C™. The answer is that we will support it to the extend that is has a 6809 CPU. After all that is what we (68 Micro Journal) are all about. Even had a call from a reader and article contributor informing me that he was nearly complete on a project that expands his 80C to a full 32K RAM, support bus (talk to the outside world), disk interface and patch of f the more popular *Standard S50 bus disk operating system.

* You might note in my references to the S50 bus I have begin to preface it with 'Standard'. I get calls daily from potential users wanting information concerning the 68XX series, running on a 'standard' bus or backplane. It's about time that we all realize, and refer to it accordingly, that the 50 pin bus most of us hang our boards on is the 'Standard'.

THE MC6809– Processor for the 80s

TIM AHERN & JACK BROWNE
Applications Engineers
Motorola Semiconductor Products, Inc.
3501 Ed Bluestein Blvd.
Austin, Texas 78721

The M6809 microprocessor unit (MPU) is the third generation addition to the M6800 family of microprocessors. The MC6809, introduced in late 1978, was designed by a highly trained team of over 100 people. It has the major architectural features required to make the M6809 the ideal choice for high level language (HLL) execution or standard controller applications.

The MC6800, originally introduced in 1974, was designed primarily to replace discrete logic, consequently its data manipulation capabilities were somewhat limited.

The MC68ATX MPU was the second generation member of the M6800 family. In addition to serving as a stand alone MPU, the MC68ATX is the CPU used in the MC6801 family of processors. It enlarged the M6800 instruction set with the 17 new or modified instructions listed below, while retaining object code compatability. Another key feature allows concatenation of the A and B accumulators to form a 16-bit wide double accumulator.

ABX	Add B to X
ADDD	Add Double
ASLD	Shift Left Double
BHS	Branch if Higher or Same
BLO	Branch if Lower
BRN	Branch Never
LDD	Load Double
LSL	Logical Shift Left
LSLA	Logical Shift Left A
LSLB	Logical Shift Left B
LSLD	Logical Shift Left Double
LSRD	Logical Shift Right Double
MUL	Multiply
PSHX	Push X
PULX	Pull X
STD	Store Double
SUBD	Subtract Double

The MC68ATX also has an additional IRQ interrupt input, IRQ2. Timings for key instructions were also reduced to optimize execution time. In fact, the MC68ATX executing a piece of typical MC6800 code actually reduces execution time by an average of 20%!

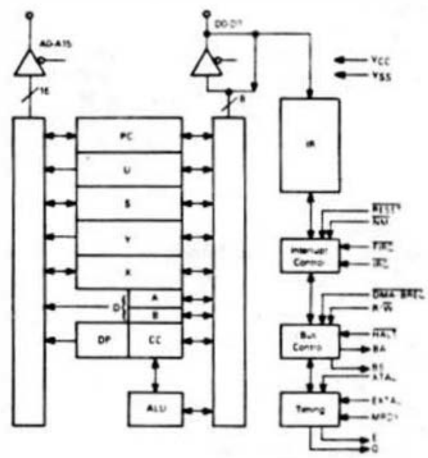
The M6809 design team benefited from the experience of the previous 8-bit MPU designs. Since the MC68000 was being designed concurrently with the MC6809, separate design teams were employed for each device. Because of this, the M6809 designers were allowed to focus on problems unique to 8-bit MPUs rather than compromise the 8-bit design to accomodate 16-bit requirements. This ability to concentrate solely on 8-bit requirements led to the major hardware and software innovations which qualify the M6809 as the best 8-bit MPU!

There are two versions of the M6809 available today: the MC6809, with an on-board oscillator; and the MC6809E, requiring an external clock generator. Both versions are available in three bus speed ranges - 1 Mhz, 1.5 Mhz, and 2 Mhz.

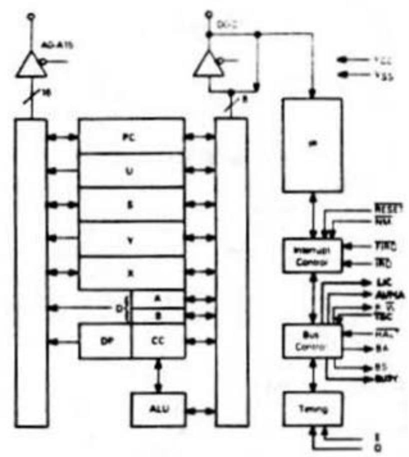
Note: Throughout this article, the term M6809 will be used as a generic reference. The two versions will be differentiated by part numbers: MC6809 refers to the on-chip oscillator version, while the MC6809E refers to the version which requires an external clock generator.

Block diagrams for both versions, shown in Figure 1, reveal that the primary differences are in the Bus Control and Timing circuits.

Block Diagrams



MC6809



MC6809E

Figure 1

The MC6809E design started after first silicon was available for the MC6809.

The chip layout remained exactly the same, but the Bus Control and Timing Circuits of the MC6809 were removed and replaced with the circuits necessary for the MC6809E. The common M6809 CPU kernel consists of the register set shown in Figure 2, an ALU, Instruction Register, Interrupt Logic, Address and Data bus logic and buffers.

Register Set

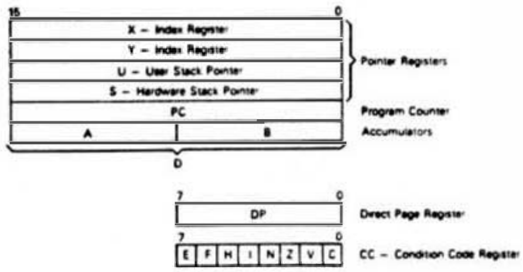


Figure 2

The register set of the M6809 is a superset of Motorola's existing 8-bit MPU's. Three registers were added to the register set of the original MC6800 - a Direct Page Register, a User Stack Pointer, and a second Index register. There are two 8-bit accumulator registers - the A & B registers which are used for data manipulation and serve as holding registers for arithmetic calculations. The M6809 has many 16-bit type arithmetic operands including shifts, loads, stores, and an 8 x 8 multiply. The 16-bit shift, load and store operations use both accumulators - with the A register treated as the most significant byte. When the A and B register are concatenated, they are refered to as the D register.

The Direct Page Register (DP) is one of the new registers. The contents of this register form the high order byte of the address bus during instructions utilizing the Direct Addressing mode. This register may be changed to allow direct addressing anywhere in the 64k memory map as opposed to the MC6800 which only allowed direct addressing in the first 256 bytes of the memory map. Direct addressing uses the immediate byte of the instruction as a one-byte pointer into a single 256-byte "page" of memory. This shortens instruction execution time as the Most Significant Byte (MSB) is furnished by the Direct Page Register. MC6800 compatability is ensured, as a Reset clears the Direct Page Register.

The M6809 has four 16-bit pointer registers available to the user. The U and S registers support stack oriented instructions such as PSH and PUL. The S register is used as the hardware stack pointer to support interrupts and subroutine calls. The U register gives the designer the capability of maintaining an independent stack. The other two registers, X and Y, are registers intended primarily for use as Index Registers, although special indexing modes allow them to be used to maintain additional stack areas. All four pointer registers may be used as Index registers allowing Index Addressing, Indirect Addressing or Indexed Indirect Addressing. These pointer register capabilities allow the M6809 to function efficently as a stack processor, allowing the MPU to support high level languages and modular programming techniques.

The MPU's program counter, while primarily utilized by the processor to address the next instruction to be executed, may be used like an index register, thus allowing addressing relative to the Program Counter.

The Condition Code Register defines the state of the MPU such that conditional branch instructions may be used. The condition code register also allows masking of certain interrupts.

This set of registers is manipulated with a set of 59 instructions. 1464 different opcodes are available to the programmer if all modes of the instructions are considered. However, only the 59 mnemonics must be remembered when using the Macro Assembler as it picks the applicable opcodes.

Software costs are rising so fast that in many systems, the hardware costs are insignificant. The M6809 was designed for ease of software development. Very effcient Position Independent Code (PIC) may be written using the capabilities of the M6809. The program counter may be used as a pointer to provide offsets within the program. For example: When a piece of PIC is executed, the stack addresses, peripherial addresses, and other addresses may be specified as offsets from the current PC address. Other key factors in effective position independent code writing are the use of long and short relative branch instruction and the Load Effective Address instruction. The relative branch instructions allow Program Counter Relative branching. When an 8-bit offset is used, control may be transferred anywhere within a 256 byte area. A 16-bit offset allows transfer of control anywhere in the entire 64k address space. The following are examples of the relative branch instruction.

```
DECA          Decrement A Accumulator
BEQ CAT       If A = 0 then goto CAT
              (CAT is within +/- 128
              bytes)

INCA          Increment A Accumulator
LBEQ BOWSER   If A = 0 then goto BOWSER
              (BOWSER is within +/-
              32,768 bytes)
```

The Load Effective Address (LEA) instructions work by calculating the effective address of an indexed instruction and storing it in the specified pointer register. This allows

the designer to utilize all the internal addressing hardware associated with the MPU. Below are some examples of the LEA instructions. With these key instructions, a software designer may generate efficient Position Independent Code.

Instruction	Operation
LEAX 10,X	X + 10 --> X
LEAY A,Y	Y + A --> Y
LEAX D,Y	Y + D --> X
LEAU -10,U	U - 10 --> U
LEAX TABLE,PCR	See text

Note how the registers may be incremented or decremented using the LEA instructions. In addition, registers may be used as offsets as shown above. The Program Counter may be used as a pointer register with 8 or 16-bit signed offsets. As in Relative Addressing, the offset is added to the current PC to create the Effective Address. The last example calculates the offset of TABLE and adds it to the current value of the PC. This value is then placed into the X register. Tables related to a particular routine will maintain the same relationship after the routine is moved, since addresses are calculated when the code is executed.

As seen by the hardware designer, the M6809 has sixteen address lines, eight data lines, three interrupt inputs, bus control and timing signals. The bus control and timing signals are different for each version of the M6809 MPUs.

Common Bus Control signals in the two different M6809 versions are HALT, R/W, BA, and BS. HALT is used to remove the M6809 MPU from the bus to allow DMA or multi-procesor operations. R/W tells the system whether the MPU is doing a read or a write operation. BA and BS allow the system to monitor the MPU operation states.

Six signals are unique to the MC6809 Bus Control and Timing circuits. All timing is generated by a parallel resonant crystal connected to the Extal and XTAL pins. In addition to the crystal, two additional capacitors (27 pf with 4 MHz crystal) are required to prevent harmonic oscillations. The crystal oscillator feeds a divider network which produces two system clocks, E and Q, which run at one fourth the crystal frequency. E is the standard M6800 bus timing signal also referred to as Ø2. Q is a clock which leads E by one quarter bus cycle. The operation of E and Q is shown in Figure 3.

Bus Timing Signals

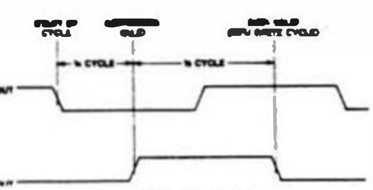


Figure 3

The rising edge of Q may be used by the system as an address valid strobe, and the falling edge may be used as a data valid strobe on a write cycle. The falling edge of E is used by the MPU to latch data during read cycles. M6800 peripherials also use this falling edge to latch data from the MPU during write cycles. E and Q provide multiple clock edges which can be useful in the generation of RAS and CAS signals for dynamic RAM.

If slow speed memories are to be incorporated in the system, the clocks may be stretched up to 10 microseconds by pulling MRDY low. This signal will stretch E high and Q low until released.

DMA/BREQ is an active low input which allows another bus master i.e. DMA controller, RAM refresh controller, or co-processor to acquire the buses.

The MC6809E has six multiprocessor control and timing signals. Bus timing for the MC6809E is also controlled by E and Q except that they are inputs from an external clock generator. Q is a TTL compatible input whereas E is a MOS level type input. The MOS level input circuitry minimizes the skew between the external clock generator and the internal MPU circuits. Notice that E may be driven with a pullup resistor. Figure 4 shows a sample clock generator.

Sample Clock Generator

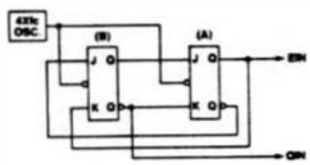


Figure 4

Three State Control (TSC) allows the designer to control the Addresses, Data bus, and R/W line on a cycle-by-cycle basis, whereas HALT can only stop after execution of an instruction.

AVMA is the Advanced Valid Memory Access signal indicating that the MPU will use the bus during the next cycle. AVMA goes low during HALT or SYNC states to guarantee the systems' data integrity.

The BUSY output provides the indivisible memory operation required for a "test-and-set" operation. Operations of this type are required for efficient multiprocessor support on a common bus.

LIC indicates that the first byte of an opcode will be latched at the end of the preaent bus cycle.

This choice of MPUs allows the designer to choose the optimum MPU version to meet his system requirements. Several examples demonstrate these features. For example - an onboard oscillator with external crystal provides all system clocks. For alower ROMS, MRDY may be used to extend access times. A schmidt trigger reset circuit allows the use of minimal external components. A resistor/capacitor

combination is all that is required for a power-on Reset circuit. Add a Reset switch, and the installation is complete.

One of the nicest system aspects is that of bus loading. Many times simple systems have become less simple due to minimal CPU loading characteristics. The M6809 was made with the circuit designer in mind. All signal lines (clocks, address, data, and control lines) are supplied with enough drive for 4 LS TTL loads.

As stated before, the M6809 has provisions for three separate levels of interrupts. One of which is a Non-Maskable Interrupt (NMI), which cannot be masked under software control. The NMI is useful in power-down situations, and real-time interrupt servicing. The other two interrupts are maskable under software control. One is "faster" than the other in that a response to a Fast Interrupt Request (FIRQ), stacks only the Condition Code register and the Program Counter. As can be seen, the M6809 is well suited for interrupt operation. Some popular microprocessors take even more time to recognize an interrupt than it takes the M6809 to recognize and stack its entire register complement! Three levels of software interrupts are also provided.

The M6809 provides the user with the capability of vectoring by device. This is accomplished by a control line which signifies Interrupt Acknowledge (IACK). When this line goes true, it signifies that the next two bus cycles will be a vector fetch, and that, if desired, the user may supply his own vectors at this time. During this vector fetching, the user must turn off the highest page ROM, lest the vectors be read from it. The interrupt vectors and their addresses are listed below.

RESET	FFFE
NMI	FFFC
SWI	FFFA
IRQ	FFF8
FIRQ	FFF6
SWI2	FFF4
SWI3	FFF2
RESERVED	FFF0

Interrupt Vectors

The control line from which IACK is derived is actually one of a full set of 4 MPU states. They are as follows:

BA	BS	MPU STATE
0	0	NORMAL (RUNNING)
0	1	SYNC ACKNOWLEDGE
1	0	INTERRUPT ACKNOWLEDGE
1	1	HALT OR BUS GRANT

As can be seen, the two control lines - Bus Available (BA), and Bus Status (BS) may be decoded to provide the user with the internal state of the M6809. The other two states which have not been mentioned will be discussed now.

Sync Acknowledge: This signal is in response to the SYNC software command which allows hardware synchronization to a software program. As the program executes the SYNC instruction, all program execution stops and waits for a hardware interrupt. If an interrupt occurs, and its associated mask bit is set, then the program will continue execution. If the mask is clear, the program will fetch the interrupt vector and service the interrupt before continuing on in the program. Of course, the NMI can not be masked, and will be serviced before falling through the SYNC command. This instruction is similiar to the old Wait for Interrupt (WAI) instruction on the MC6800, but with obvious improvements.

Halt/Bus Grant: This condition exists when the processor has entered a halt condition via a pulling low of the HALT or DMA/BREQ line. As shown, these hardware properties are well suited for an advanced software machine such as the M6809.

The M6809 as a Controller

The MC6809 easily adapts to a controller-type environment by using the multitude of M6800 series of peripherals which are totally compatible. Because the M6809 converses with peripherals in an efficient memory-mapped configuration, no special I/O instructions are required, thus keeping the instruction set more regular and consistant. Figure 5 an example of how an MC6809 may be placed in a typical controller application.

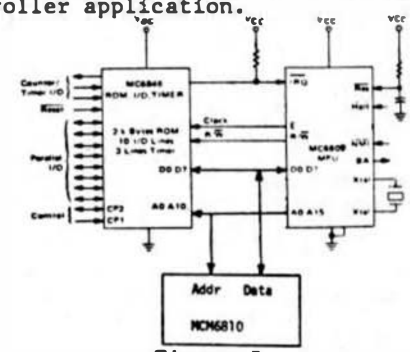


Figure 5

This application is shown using a total of 3 parts - the MC6809, a MC6846, and a MCM6810 or some other type of RAM.

The MC6846 contains two kilobytes of mask-programmed ROM, an 8-bit parallel I/O port, and a 16-bit timer. Since many controller applications require aeveral variations with different programs, the cost of mask-programmed ROMS may be too high to be justified. Figure 6 shows a system in which the MC6809 uses standard EPROMS for program storage. Although more parts are required, a greater degree of system flexibility may be attained through this design than by using a single-chip microcomputer such as the M6801 or M6805 family of MPUs.

The basic controller can be embellished by the addition of any of the many available M6800 peripherals. The question is, where does a controller end, and a smart system begin? Of course, all controllers are systems, but, just for the sake of arguement, we

will call a system one which has extended interaction with humans, such as would be required in a personal computer or small business system.

The Expanded System

Several companies have chosen to implement the M6809 in products whose end functions range from low cost "color computers", to extended personal computers, and on to even higher sophisticated business systems. The low cost systems are basically one step up from a controller design, with a minimal number of "bells and whistles", while both the personal computer and small business machines have the capbility of greater expansion e.g. more memory - > 500k bytes with memory management unit (MMU) extensions, and provisions for higher level languages such as Pascal and BASIC09. BASIC09 is a sophisticated programming language system that is a leap in state-of-the-art microcomputer system software. Its many advanced features are aimed toward efficient, structured software development and testing. BASIC09 is the result of an intensive, two-year development project, and is quite possibly the most sophisticated general-purpose microcomputer program ever written.

BASIC09 may use multiple, independent named procedures in memory simultaneously which are re-entrant, position independent and ROMable. Procedures have local variables, are called by name and pass parameters to others. It includes enhanced I/O capabilities, and has compiler performance; an integrated three-pass compiler and interpreter design. In addition to its powerful built-in data structures, BASIC09 allows user-definable "record-like" structures - a powerful Pascal-like feature. Included are the full complement of math and transcendental functions, and a complete editor-debugger.

In addition to software developed with Motorola, one software company has implemented an operating system for the M6809 which has almost as much power as the UNIX operating system developed by Bell Labs. Its syntax and operation parallel that of UNIX. These personal computers can be enhanced to provide the user with hardware and software development systems whose price had previously been too high for other than large companies. These systems are now reasonably priced, and any serious hobbyist could easily justify one.

Larger... and Beyond

With the advent of the MC6809E (external clock version), larger multi-processor systems have become easier to design as shown in Figure 6. Because all clocks are externally generated, synchronization to peripherials such as CRT controllers and other MPUs have become a snap to implement.

With the new MC6829 Memory Management Unit (MMU), the M6809's memory map has been expanded to 2 megabytes with built-in provisions for multi-user and multi-tasking environments. The principal function of the MC6829 Memory Management unit is to expand the address space of the M6809 from 64k bytes to a maximum of 2 Megabytes. Each MMU is capable of handling four different concurrent tasks, including DMA. The MMU can also protect the address space of one task from modification by another task. Memory address expansion is accomplished by applying the upper five address lines of the processor A11-A15 along with the contents of a five-bit

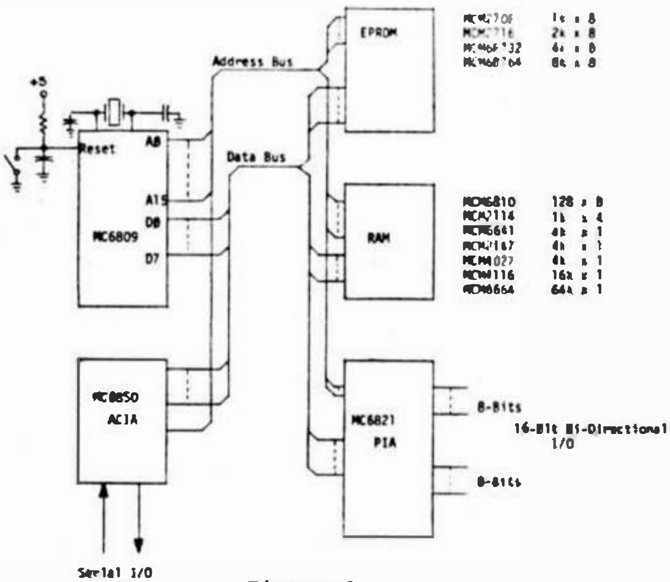


Figure 6

task register to an internal high-speed mapping RAM. The MMU output consists of ten physical address lines (PA11 - PA20), which, when combined with the eleven lower address lines of the processor (A0 - A10) forms a physical address space of 2 Mbytes. Each task is assigned memory in increments of 2k bytes up to a total of 64k bytes. In this manner, the address spaces of different tasks can be kept separate from one another. See Figure 7.

The resulting simplification of the address space programming model will increase the software reliability of a complex microprocessor system.

Another significant addition to the M6809's complement of support chips is the MC6839 Floating Point Rom. This ROM is totally position independent - it can be placed anywhere in the memory map.

The MC6839 PR implements the Floating Point Standards currently proposed by the IEEE. No absolute RAM is required as all operands are carried in registers or on the stack. This allows re-entrant code and provides the support required by high level languages such as Pascal. Single, double, and double extended formats are provided allowing numbers ranging to 64 digits with 16 digit exponents. The FPR supports the operations listed in the following table.

- Add
- Subtract
- Multiply
- Divide
- Remainder
- Square Root
- Integer Part
- Negate
- Compare
- Convert Integer to Floating Point
- Convert Binary to Decimal

The MC6839 Floating Point Rom represents Motorola's first entry into the Standard Product Rom (SPR) marketplace.

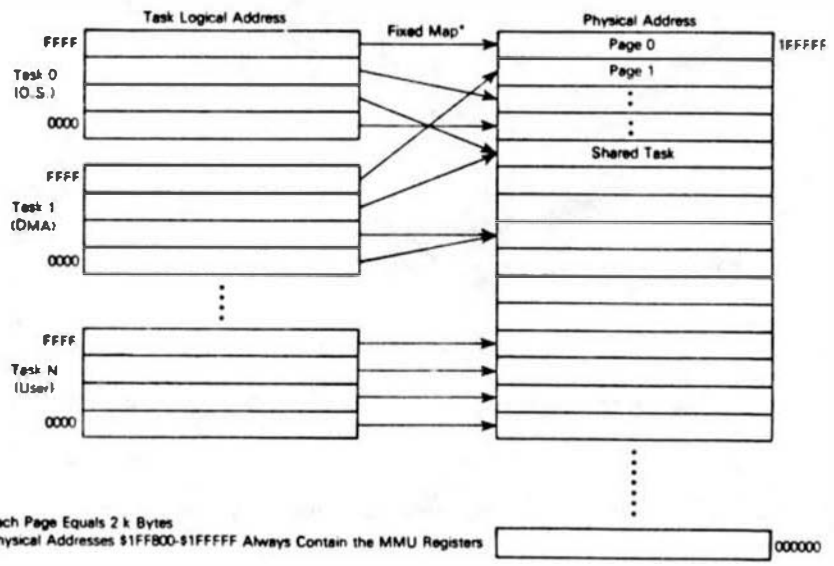
SPRs will contain modular position independent code, thus freeing designers from the need to generate individual sets of common software routines. This market has come of age because of the ease of writing software which is totally transportable. This transportability is due to Position Indepedent Code which can be effectively and easily written for the M6809 and M68000 families of processors.

The MC6842 Serial Direct Memory Access Processor (SDMA) is but another entry into the M6800-M6809 bus compatible peripherial market. The MC6842 provides a high speed serial link between microprocessors or intelligent controllers in distributed processing systems. Using IBM's Synchronous Data Link Control (SDLC) protocol, the MC6842 is capable of handling multidrop, point-to-point, or loop configurations. Many HDLC protocol features are also supported.

The SDMA processor accepts commands from the local microprocessor to either transfer data or issue link-level commands. The SDMA issues and responds to most link-level commands, ensures data integrity and validation, and handles some error recovery.

Considering all available microprocessors, the M6809 family represents a consistant choice of 8-bit micros in all segments of the representative markets i.e. controller environments, small business systems, and the ever-present home computer market.

LOGICAL TO PHYSICAL ADDRESS MAPPING EXAMPLES



*Each Page Equals 2 k Bytes
Physical Addresses $1FF800-$1FFFFF Always Contain the MMU Registers

Figure 7

DYNAMITE

I've recently had the pleasure of using Dynamite, a disassembler from Computer Systems Center, 13461 Olive Blvd., Chesterfield MO 63017. Dynamite was written by Philip Lucido. Philip is obviously very familiar with FLEX. He has made using his disassembler as nearly as possible like using the assembler. One specifies options by appending '+' followed by a series of single letter option specifiers. I for one, appreciate being able to specify everything for a run in the command line. I don't like to play question and answer, particularly if the program is going to run a while and then ask me for more input. That way, I can't go get a bottle of Vernor's (ginger ale) while the program is running. Dynamite allows the user to set things up so it can run unattended.

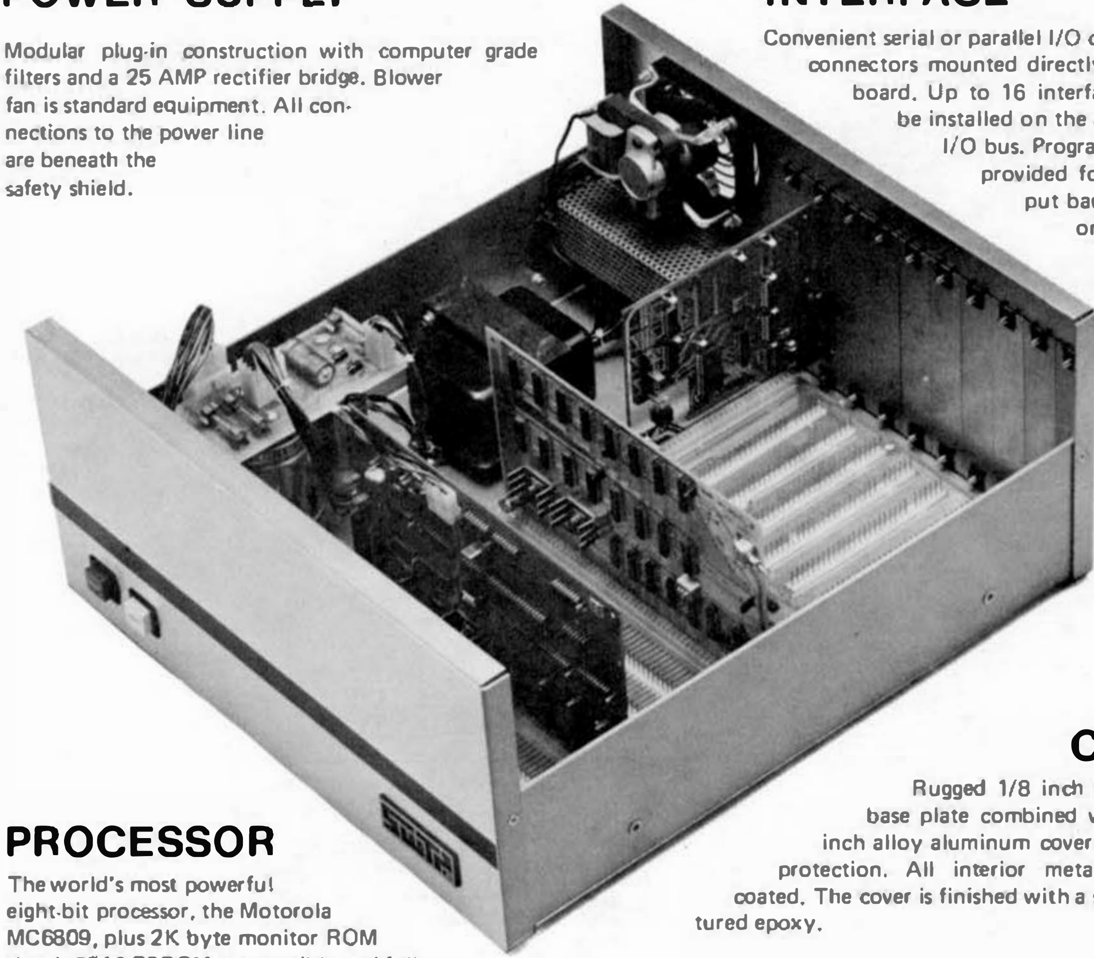
WE HAVE A 6809 FOR YOU

POWER SUPPLY

Modular plug-in construction with computer grade filters and a 25 AMP rectifier bridge. Blower fan is standard equipment. All connections to the power line are beneath the safety shield.

INTERFACE

Convenient serial or parallel I/O cards have DB-25 connectors mounted directly on the circuit board. Up to 16 interface devices may be installed on the address decoded I/O bus. Programming strips are provided for input and output baud rate selection on each port. All outputs are fully buffered.



CABINET

Rugged 1/8 inch alloy aluminum base plate combined with a solid 1/8 inch alloy aluminum cover for unsurpassed protection. All interior metal is conversion coated. The cover is finished with a super tough textured epoxy.

PROCESSOR

The world's most powerful eight-bit processor, the Motorola MC6809, plus 2K byte monitor ROM that is 2716 EPROM compatible and full buffering on all output lines. Built-in multiuser capability, just add I/O cards to operate a multi-terminal system.

MEMORY— You can purchase the computer with either 8K bytes of RAM memory (expandable to 56K), or with the full 56K. The efficient, cool running dynamic memory used in this system is designed and manufactured for us by "Motorola Memory Systems Inc."

PERIPHERALS—The wide range of peripheral hardware that is supported by the 6809 includes: dot matrix printers (both 80 and 132 column), IBM Electronic 50 typewriter, daisy wheel printers, 5-inch floppy disk system, 8-inch floppy disk systems and a 16 megabyte hard disk.

SOFTWARE— The amount of software support available for the 6809 is incredible when you consider that it was first introduced in June, 1979. In addition to the FLEX9 operating system, we have a Text Editor, Mnemonic Assembler, Debug, Sort-Merge, BASIC, Extended BASIC, MultiUser BASIC, FORTRAN, PASCAL and PILOT.

69/K Computer Kit with 8K bytes of memory . . . . $ 495.00
69/A Assembled Computer with 8K bytes of memory . . . . $ 595.00
69/56 Assembled Computer with 56K bytes of memory $1,595.00

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

6809 DISK SYSTEMS

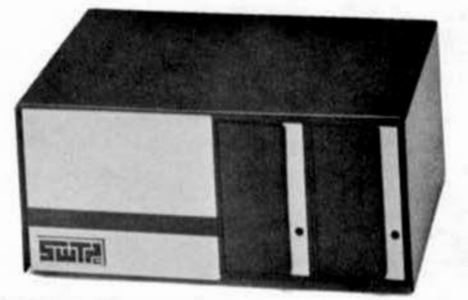
All disk systems are supplied with our version of FLEX 9, the world standard disk operating system for the 6809. Our systems normally operate in double density format, but they are compatible with single density, or single sided recording formats. FLEX is supplied with over forty utilities, many of which are only available with our systems.

Our disk systems offer you mass storage at low cost. The cost per thousand bytes of storage for our various systems is shown in the chart. Other 6809 disk systems have costs up to three times greater for the same general type drive.

TYPE	CAPACITY	COST
D-5	720,000 bytes	$1.80 per/K
DT-5	1,400,000 bytes	$1.16 per/K
DMF-2	2,400,000 bytes	$1.04 per/K
CDS-1	16,000,000 bytes	$ .27 per/K

D-5 Two double sided, double density, 5" disk drives with a total on line capacity of 720,000 bytes of data. Includes cabinet, power supply, connecting cable and controller. Controller will operate up to four drives. This is an ideal disk system for small stand alone word processing systems, or for businesses that do not work with large inventories.

14 x 6 x 10 — 20 lbs $1,295.00



D-5 or DT-5

DT-5 Double track density version of the D-5. The DT-5 uses two 96 track per inch drives to provide an on line capacity of 1,400,000 bytes. Includes cabinet, power supply, connecting cable and controller. Controller will operate up to four drives. This is a disk system with enough capacity to include small inventories of up to 1,000 items, plus the usual business package of general ledger payroll, etc.

14 x 6 x 10 — 20 lbs $1,695.00

DMF2

DMF-2 Double sided, double density, dual eight-inch disk system with an on line capacity of 2,400,000 bytes. Our "top of the line" disk system features a DMA type controller for fastest possible data transfers. This drive was designed for larger businesses and multi user installations. The DMF-2 will provide the fast operation necessary for systems running multiterminals under the UniFLEX operating system. Complete with a heavy duty 1/8-inch metal cabinet, power supply, connecting cable and controller. The controller will operate up to four drives.

17½ x 5 x 21½ — 53 lbs $2,495.00

CDS-1 This "Winchester" type hard disk provides both large storage capacity and high speed operation. The CDS-1 is the answer for systems that must handle large inventories or systems with more than four terminals. The controller has its own processor and uses DMA data transfer.

CDS-1 — 115 lbs $4,395.00

CDS-1

SwTPc

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

I've had four disassemblers at this point. The first was published in one of the very early issues of Kilobaud. It was a start, but not much more. The same could be said of the TSC disassembler. Ed Smith's Software Works disassembler was a major improvement over the earlier ones. Ed called it a Source Code Generator. The improvement came in a couple of forms. First of all, the Source Code Generator generates labels. They are formed as the letter L followed by a number. The numbers start at 0001 and increment by 1, being assigned as labels are needed in the program. Ed made provision for you to disassemble a program from memory to your terminal or printer, or to a disk file. When you disassemble to a terminal, you will find areas of most programs in which the disassembler gets confused and disassembles certain operation codes, (or rather what it thinks should be op-codes) as '***'. These are such areas as strings, jump tables, etc. When you look at the disassembled code to the screen you decide what type of 'data' area you are looking at, and jot down on a scratch pad the address limits of these areas. When the disassembler is run again, you are prompted for the limits and types of these areas, and you may enter them and run the disassembler again, checking to see if the listing now makes more sense. This process is repeated until you are satisfied with the output, and you may then specify saving the output to a disk file that may later be used to assemble the program again.

Dynamite goes a couple of steps beyond this. It also needs your help in determining what areas contain data, jump addresses, etc. Dynamite allows you to enter these areas limit addresses each time, or to create a text file that contains the information. It will then read the text file and use the information. Dynamite, unlike the earlier disassemblers never will give you '***' for an op-code. If all else fails, it will resort to FCB $29 etc. For this reason, even if the disassembler output doesn't make sense, it will always assemble to produce the original program.

Dynamite runs under FLEX9 with the 6809 processor, but will disassemble either 6809 or 6800 code. It is eminently useful for disassembling 6800 utilities so you can change the FLEX equates and reassemble for your 6809 system. The external equates, ie. address references to areas outside of the limits of the program, always appear at the beginning of the disassembled source, and the equates thus generated may be changed very easily. There is one additional feature of Dynamite. It comes with a set of data files containing the Equates for normal entry points to various operating systems and monitors. To be more specific, there is a file of equates for FLEX2, FLEX9, MINIFLEX, SWTBUG, and SBUG-E. You may use, in addition to your command file of data area addresses, one of these files, and a label file of your own. Thus as you proceed with the disassembly of a program, you can, as understanding comes, define meaningful labels and create a file of them. You can modify your command file to include new areas of data or strings as they are discovered, and eventually arrive at a reasonably disassembled file. It is much easier to edit a small data file repeatedly, than to edit a large disk file, and you need never prepare a final output source file until you are satisfied with the results to your screen.

Dynamite works from disk file to screen or another disk file. It is never necessary to have the program in memory durning the process. One of the very nice features, is an option that allows the display of the ASCII equivalents of each instruction in the disassembly listing to the CRT. This makes the text strings stick out like sore thumbs, and it is quite easy to determine their limits to be entered in the Command file for the next pass of the disassembler.

I was most impressed by the fact that the disassembly to the CRT is identical in format and content to the output of the TSC assembler. In fact, a run of the disassembler to the printer is nearly indistinguishable from a run of the Assembler on the source file generated on a disk. The software is entirely compatible with the P.CMD of FLEX for output to a printer. It correctly manipulates SWITCH so that prompts appear on the CRT and listing goes to the printer.

In summary, Dynamite is not just another disassembler, but a better approach that is a major improvement in the capabilities of such software. I must give it an excellent rating, and recommend it as the best one I have seen to date.

Dynamite may be ordered from: Computer Systems Center, 13461 Oliver Blvd., Chesterfield, MO 63017, telephone (314) 576-5020.

See advertisement this issue.

Ron Anderson —

zingg for SWTPC cassette

FASTER CASSETTE OPERATION WITH THE SWTP 6800 SYSTEM

by A.J. Hall, Ultrasonic Technology, Queen Mother's Hospital, Glasgow.

Introduction

Commercially supplied software for cassette based systems is usually supplied in the Kansas city format recorded at 300 baud using a 2400 Hz marking carrier which varies according to whether binary "1"s or "0"s are being recorded. A binary "1" is represented as 8 cycles of frequency 2400 Hz, while a binary "0" is represented as 4 cycles of 1200 Hz. When data is retrieved from the tape during playback, the marking carrier is detected and processed, producing a clock for data detection. This self clocking mode of operation is tolerant to the frequency variations arising from tape speed fluctuations as the tape passes the playback head; these fluctuations are caused by variations in drive motor speed, slipping clutch friction etc. and are found, to a greater or lesser extent, in all cassette recorders.

The AC30 cassette interface which is the only one offered by South-western Technical Products can control two cassette recorders independently and is designed to the Kansas city standard which is painfully slow when it comes to loading large programs such as an 8k Basic or a large Editor Assembler. A review of alternatives for faster cassette operation by P.A. Stark (1) indicates that there are a number of hardware and software solutions. However having purchased the AC30 cassette interface one is reluctant to discard it for an alternative, possibly non standard approach. Therefore, methods of increasing the speed of operation in a simple straightforward manner were considered. It was thought that for program development and the loading of short programs, a doubling of the baud rate to 600 would be a useful improvement, especially if only minimal hardware modification is needed.

The ASCII format used to record and playback data effectively doubles the time taken for recording or playback as, when a SWATBUG "D" (DUMP) is executed, the 8 bit binary word at each saved memory location is formatted into two ASCII characters, as shown in Fig. 1 (a fuller description of the format is given in reference 2). Similarly, when loading from tape, the binary word for each memory location is formed from two ASCII characters. The redundancy in the hardware (the number of cycles used to record each byte and the frequency of recording) and software (two ASCII characters per byte) means that programs such as a 12k Assembler or similar sized Basic take

some 15 minutes to load and this can be most frustrating if a system crash occurs during use. The redundancy in the software loading format can be reduced by using a commercially available binary loader and the implications of this and modifications to the AC30 hardware to operate at a higher baud rate are considered below:

(A) Software Modification

The load time can be reduced using a binary loader which dispenses with the previously described ASCII format. A commercial example is Ed. Smiths Binary loader (V2A.2.0) which has both a save and a load routine. When saving a program the loader generates a short SWATBUG* formatted program in ASCII at the beginning of the tape and then writes to tape, in binary format, the selected contiguous area of memory. To load such a tape the standard SWATBUG* "L" routine is used, this loads the short ASCII formatted loader program which in turn automatically loads the binary formatted tape. When using a binary loader the "read status" toggle switch on the AC30 must be set to the "on" position until after the "S" sign followed by a "G" appears, it can then be returned to the centre position to allow the tape to stop automatically under program control.

As supplied, Ed Smiths binary loader is located from $1C00-$1DE2 which is in the area of memory used for our assembler or basic. The first step is to relocate it to be just above the end of the program to be saved; easily done using Ed Smiths Eprom support program as the loader is directly relocatable. The program to be saved is then loaded in and the binary loader save routine is used to dump the program to tape - possibly at 600 baud if you decide to implement the hardware changes discussed below.

(B) Hardware Modification

The Kansas city format specifies that a binary "0" is represented by 4 cycles of 1200 Hz while a binary "1" is represented by 8 cycles of 2400 Hz, this latter frequency is well below the upper 3 db bandwidth of a reasonable quality cassette recorder, for example the upper 3 db bandwidth of the Sony TC207 is 8000 Hz, thus the frequencies used can easily be doubled to 2400 and 4800 Hz respectively without changing the recording format. This is effected by altering the baud rate to 600b on both the port 1 interface card and the keyboard console. For convenience, it is suggested the modification shown in Fig. 2 is implemented as the fitting of a dual-in-line switch to the interface card, accessible via one of the large holes in the main frame, allows the baud rate to be easily altered.

In addition, a baud rate switch must be fitted to the AC30 cassette interface, and while the setting of this switch is immaterial during recording it must be set to the appropriate baud rate during playback.

The AC30 modulator/demodulator circuit diagram (to be found near the end of the AC30 handbook) and a somewhat terse circuit description (page 20 of the handbook) indicate that the circuitry is mostly logic based. The recording circuitry, using the "data in" and "clock in" signals to produce an audio tone, consists entirely of logic whose output is passed through a shaping network to produce sine waves; the output of this network is less, but only marginally so than that obtained at 300 baud. To playback tapes recorded at 600 baud the playback timing circuitry must be altered, but only minimally.

The timing circuit (see Fig. 3) made up of R15, R16, C8 and transistor Q1 differentiates between low and high frequency tones according to whether the charging voltage across C8 exceeds the threshold level of IC?? before being discharged by the switching on of Q1 - a low frequency tone allows C8 voltage to exceed the threshold while a high frequency tone does not.

This time constant formed by R16, R15 and C8 must be halved to allow for the higher frequencies used in 600 baud operation, this is simply done by inserting a switch (S1 in Fig. 3) which puts additional resistors (Ra and Rb) in parallel with R16 and R15. The new variable resistor can be adjusted either by using one of the test programs given in the AC30 manual or by measuring the existing resistance of R15 + R16 and then switching in and adjusting the new variable resistance to give half the previously measured value.

Conclusions

The changes discussed above have proved reliable and significantly improved the utility of a cassette based system. It has been found that operating at 600 baud in ASCII format is acceptable for loading short programs such as those developed using an editor assembler package which subsequently uses a linking loader to join all the program modules together. Such a program when finally complete can then be saved and loaded in binary format at 600 baud in a similar manner to that advocated for the loading of a large Basic or Editor Assembler package. If a large program is loaded at 600 baud using the binary format, the load time will be approximately one quarter of that taken to load at 300 baud in the ASCII format.

References

(1) Stark, P.A. (1979) Thoughts on the SWTP Computer System Kilobaud Microcomputing V8 pp 46-51.

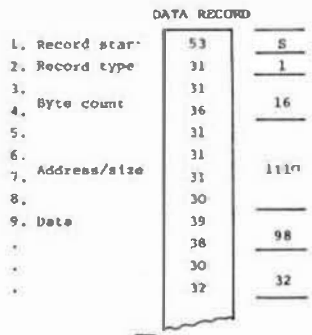
(2) Wiles, M. and Felix, A. Engineering Note 100, MCM 6830L7 Mikbug/Minibug Rom. Motorola Semiconductor Products.

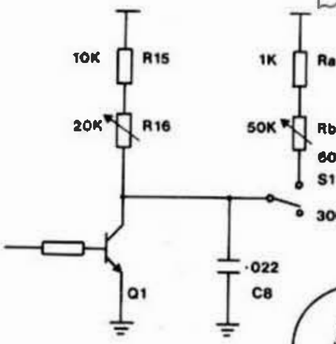
Captions

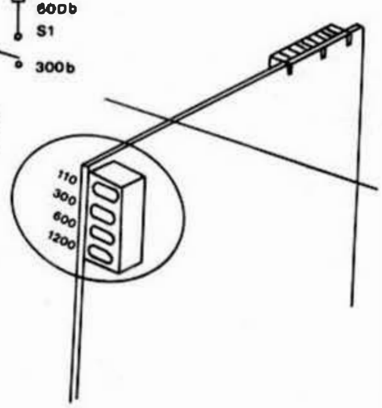
Figure 1 Recording format using two ASCII characters for each 8 bit word.

Figure 2 Dual-in-line switch fitted to interface card allows the baud rate to be changed via a rear panel access hole.

Figure 3 Modification to AC30 demodulator circuitry for 600 baud operation - Ra and Rb are added in parallel with existing R15 and R16 via new switch S1.







BIT Bucket

FOR: Smoke Signal Broadcasting
31336 Via Colinas
Westlake Village, CA 91363

NewsRelease

LaMonto Marketing Communications, Inc., 609 Deep Valley Drive, Palos Verdes, CA 90274, phone (213) 377-67...

FOR IMMEDIATE RELEASE — **CONTACT: Jeff Swartz (213) 377-6703**

"OCTO-DENSITY" 5 1/4" DISK DRIVES NOW AVAILABLE IN CHIEFTAIN; STORAGE CAPACITY BOOSTED TO 1.5 MEGABYTES

WESTLAKE VILLAGE, CA- November 17, 1980 ... Smoke Signal Broadcasting (SSB) has announced that the new "octo-density" 5 1/4" flexible disk drive is now available in its 6809-series of Chieftain small business computers.

The drive's double-track, double-bit, and double-sided design provides a total of one and one-half megabytes of formatted storage capacity in its standard dual-drive configuration, according to Ric Hammond, SSB president.

"With the addition of this new capability, we can now support a complete range of small business systems that will meet virtually any storage requirement or software application. It also represents a versatile complement to the 6809 microprocessor", he added.

SSB's Chieftain Model 9524 also incorporates 32K RAM, two serial ports, monitor in ROM and SSB's DOS69. The computer's 6809 microprocessor allows users to run programs in BASIC at over two and one-half times the speed of 6800-based systems.

Recording density of the drive is 5,877 BPI with 80 tracks per side. Track-to-track time is 1 milliseconds and TPI is 96.

Single quantity retail price of the Chieftain Model 9524 is $4,075 with OEM and dealer discounts available. Delivery is from stock.

For further information, contact Jim Allday of SSB at (213) 889-9340.

KENYON MICROSYSTEMS

November 25, 1980

Mr. Don Williams
'68' Micro Journal
3018 Hamill Road
P. O. Box 849
Hixson, Tennessee 37343

Dear Don:

Just a note to let you know what's going on here at K-M.

First, we want to thank all the folks who have purchased tFORTH. We especially want to thank those customers who have taken the time to write or call with nifty little patches, application programs and the like. We're trying to put these all together and hope to mail them out as a Christmas "Thank You" to those who have parted with their ever shrinking dollars to purchase the tFORTH system.

Second, we are now shipping tFORTH version 1.1. The biggest differences are three new hi-speed math words and the capability to mix unlike disk drives (i.e. 5¼" and 8", different number of tracks per disk, different number of sectors per track). Please see the addenda sheet enclosed.

Third, purchasers of tFORTH+ will now be receiving two disks in their package. One disk contains the tFORTH+ software and the other is a computer aided instruction course called "Going FORTH". This course will teach those new to the language the fundamentals of programming in FORTH. Even with these extra goodies, the price remains the same.

Last, tFORTH is now available for the Motorola Exorciser.

Thanks,

Tom

Thomas W. Kenyon
President

Conrad Swartz, PhD, MD
PO Box 1304
Iowa City, Iowa 52244

November 20, 1980

68 Micro Journal
3018 Hamill Rd
PO Box 849
Hixson, TN 37343

Gentlemen:

I'd like to recommend an exceptionally helpful and conscientious dealer— Jerry Koppel (AAA Chicago Computer Center of Wheeling, Illinois). Though busy he has helped me many times with rapid mail service and phone advice for my SS-50C system so promptly that the distance between Iowa City and Chicago has seemed negligible.

He has helped me upgrade my system by letting me trade up as my needs have grown. Unlike other dealers (especially associated with Ohio Scientific I had before) he does not send things COD without telling me first. In other words, he has given me my money's worth and has not sprung additional charges, as some other dealers have. He is very cost and value-conscious but has always reviewed quality limitations with me carefully before a purchase has been concluded.

Sincerely yours,

Conrad Swartz

Conrad Swartz, PhD, MD

O'CONNOR ASSOCIATES
CUSTOMIZED HARDWARE & SOFTWARE FOR MICRO COMPUTERS
6315 W. RAVEN ST. • CHICAGO, ILL. 60646 • 312-792-3253

Don Williams, Editor
'68' Micro Journal
3018 Hamill Rd.
Hixson, Tennessee, 37343

Dear Mr. Williams,

As a free lance programmer I have done a little bit of programming on a lot of different machines. Until recently, I thought the the Z80 was the best micro around including the 6800. Then I did an assembly language program on the new TRS-80 Color Computer. This machine uses the 6809 and it has made a believer out of me. Now that I am back working with the Z80 again it is like moving back to the Dark Ages. The Z80 just cannot compare to the 6809. When I first started using the 6809 I thought I'd never understand all the different addressing modes. Now I don't know how to do without them.

I especially liked being able to use a register as the offset when indexing instead of a constant. The only way I know to change the index offset with other machines is to use self modifying code. I also found the relative addressing modes made is extremely easy to write relocatable code. The auto increment and decrement commands are items I have only seen in 16 bit machines befor .

The TRS-80 Color Computer may not be anyway near the league of Gimix or Motorola Exorciser, but when it comes to picking out the best chip available, someone at Tandy knows what they are doing.

Sincerely

Leah O'Connor

Leah M. O'Connor

Dear Mr Williams,

As I mentioned in my first letter, I have done a little bit of programming on a lot of different machines. For the past two years I have been working almost exclusively for the Image Producers, Inc., a software house from Northbrook Illinois. I did a BASIC demo program on the Bally Arcade which was primarily used by Bally in-house. The Arcade is built around the Z-80, and although it was not much more than a toy, it had some potential as a computer. About a year ago Image negotiated a contract with Sears to produce software for the Atari. I did two of the programs currently on sale under the Sears label, "Roman Checkers" and "Oil Wells". Image Producers also sells some software under their own label and I have written game programs for the Atari and the Texas Instruments TI99/4 under the Image label. On the TI99/4 I did Wildcatting, Roman Checkers, Frame Up, Bingo Duel, and Number Hunt. My best selling program done so far is Typing Tutor for the Radio Shack TRS-80. Although it was developed at Image Producers, it is sold under the Microsoft label.

My most recent program was done (in assembly language) on a development protype of the new TRS-80 Color Computer. Although the program was finished shortly before the Color Computer was announced, I didn't see the real computer until several weeks after the announcement. The machine I worked on was not much more than a keyboard and a PC board bolted to a piece of plywood. Between the time I first saw the machine at Image, and the time I finished the product, almost every part of the system from the keyboard to the ROMs was changed at least once. Since the program I worked on has not been released yet, I cannot mention its name; it is a teaching program which will be sold as a plug-in ROM cartridge. Your readers would probably be interested in the fact that every programmer at Image who worked on the TRS-80 Color Computer agreed that the 6809 was the best microprocessor they had worked with so far.

I am currently working on a business program for the Model III TRS-80, to be written in Z-80 assembly language. There is no doubt in my mind that the 6809 can run rings around the Z-80.

In taking some graduate level courses at the University of Illinois Circle Campus, I acquired some PDP-11 assembly language experience. The assembly language of the 6809 compares quite favorably with this minicomputer's instruction set, and is far superior to 8-bit microprocessors I have worked with (6502, 8080, Z-80, 6800).

Besides programming I have also done technical writing for Radio Shack and CBC Associates and I manufacture a lowercase kit and a reset button extender for the Model I TRS-80 whih is sold by CBC Associates.

Sincerely,

Leah O'Connor

Leah O'Connor

Editor's note: The above is inserted to show where Leah 'comes from'. It is always nice for newcomers to find out some good things we have all know for years.

MR. DON WILLIAMS SR.,
'68' MICRO JOURNAL
HIXON, TENNESSEE 37343

9-29-80

AFTER SEEING A PROGRAM TO DO NEARLY THE SAME THING, I THOUGHT THE READERSHIP MAY BE INTERESTED IN THE FOLLOWING WAY OF GETTING/MODIFYING/RUNNING DISK FILES WHICH REQUIRE FLEX'S INBUFF AS PARAMETERS OR FILE SPECIFICATIONS. THIS WORKS ON A 6809 SYSTEM, BUT PROBABLY WILL ALSO WORK ON 6800 SYSTEMS AS WELL.

AS AN EXAMPLE; TO RUN 'ASMB' WITH MODIFICATIONS, TYPE THE FOLLOWING:

```
GET 0.ASMB.CMD:MON,1.FILENAME.TXT,+LSNGB
```

THEN HIT RETURN, MODIFY THE MEMORY, AND THEN USE CONTROL P TO JUMP TO THE HARDSTART ADDRESS (0000 FOR ASMB). EVERYTHING FTER 'MON,' IS PICKED UP BY THE ASMB WHEN IT EDS IT.

THIS HAS PROVEN INVALUABLE FOR DEBUGGING UTILITIES BY USING SBUG-E BREAKPOINTS. IT CAN BE US D FOR CHANGING OINTE S, COUNTERS, ETC. BEFORE RUNNING THE PROGRAM.

IT HAS ALSO COME TO MY ATTENTION THAT SOME PROGRAMMERS DO NOT REALIZE THAT FTER SOME OWER LINE HITS OR OTHE MALFUNCTIONS OF THE SYSTEM, BASIC PROGRAMS AS WELL AS OTHERS MAY BE RECOVERED BY HITTING RESET AND THEN USING THE CONTROL P TO JUMP TO THE WARMSTART ADDRESS (0003 IN BASIC AND OTHERS) AND THEN USING 'SAVE' TO WRITE THE FILE ON DISK.

ONE PROGRAM THAT I USE BEFORE TESTING/DEBUGGING NEW PROGRAMS IS CALLED 'S3F' WHICH STORES SOFTWARE INTERRUPS (HEX 3F) IN ALL OF LOWER MEMORY. THIS INIATIALIZES MEMORY IN THE SAME WAY ALL THE TIME SO THAT A PROGRAM SHOULD ALWAYS BLOW UP IN THE SAME WAY, OR STOP IF IT RUNS WILD INTO UNUSED MEMORY. IT IS ALSO USEFUL WHEN YOU WANT TO SEE WHAT IS IN MEMORY AFTER A PROGRAM HAS RUN (A '3F' IN MEMORY STICKS OUT AS AN UNUSED BYTE).
TO US , TY E: S3F:ASMB,FILENAME

RANDY LILLY N3ET
752 S. CARLDON ST.
ALLENTOWN, PA. 18103

```
S3F 9-29-80 R.L.                 9-29-80 6809 Software   PAGE   1

                           NAM   S3F 9-29-80 R.L.
                         * 6809 FLEX UTILITY FOR SWTCP COMPUTER
                         * STORE SOFTWARE INTERRUPTS IN LOWER MEMORY
                         * ALSO SHOWS WAY TO PUT VERSION NUMBER IN SOURCE
                         * WHICH 'VER.CMD' UTILITY WILL FIND

                 CC20    MEMEND EQU   $CC20
                 C003    WARMS  EQU   $CD03

C100                            ORG   $C100
C100 20    11    VER            BRA   VEREND
C102 01 20 52 2E                FCC   $01,' R.LILLY 9-29-80'
C113             VEREND

C113 8E    CC20  S3F            LDX   MEMEND
C116 86    3F                   LDA   =$3F
C118 A7    00    STORE          STA   0,X
C11A 30    1F                   LEAX  -1,X      (DEX)
C11C 26    FA                   BNE   STORE
C11E A7    00                   STA   0,X
C120 7E    C003                 JMP   WARMS

                                END   VER
0 ERROR(S)
LAST ADDRESS: C122
```

Peter Murray
P.O. Box 49302
Austin, TX 78756
June 15, 1980

Don Williams, Editor
'68' Micro Journal
3018 Hamill Road
Hixson, TN 37343

Dear Don,

I would like to thank you and your staff at '68' Micro Journal for presenting a review of JCP in the July '80 issue of '68' MJ. Dale did a fine job of covering the major aspects of JCP, and I thought it would be helpful to pass along a procedure that will further demonstrate the use of various JCP statements.

I have enclosed a listing of the text file, WORDCNT.DOC, which contains a functional description of the procedure file WORDCNT.TXT. There is also a listing and sample run of the WORDCNT procedure. Each step of the procedure is documented; however, your readers may want to refer to Dale's article for additional explanation of the JCP statements used.

May I again thank you, and express my appreciation for the excellent forum that the Journal offers the 68XX community.

Sincerely,

Peter Murray

Peter Murray

```
+++CSORT,1.TMP2.OUT,1.TMP3,+(1)6-16,+(1)6-*,(1)3-5

=== TSC SORT/MERGE V1.3 ===

 ORT RUN #1 - 100 RECORDS

100 RECORDS SORTED
KEY PADDING WAS REQUIRED
+++PDEL,1.TMP2.OUT


DELETE "1.TMP2.OUT"? Y

+++BASIC

READY

50  OPEN "0.PRINT" AS 0:C1=0:ONERROR GOTO 500
100 INPUT "ENTER FILESPEC",L1$:OPEN OLD L1$ AS 1
200 INPUT #1,L1$:PRINT #0,TAB(C1*20);L1$;
300 IF C1=3 THEN C1=0:PRINT #0:GOTO 200
400 C1=C1+1:GOTO 200
500 PRINT #0:END
RUN

ENTER FILESPEC? 1.TMP3.TXT
A  6                ADDITIONAL  1       AGAIN  1            ALONG  1
ALSO  1             AND  4              APPRECIATION  1     ARTICLE  1
ASPECTS  1          AT  1               AUSTIN  1           BE  1
BOX  1              COMMUNITY  1        CONTAINS  1         COVERING  1
DALE  2             DEAR  1             DEMONSTRATE  1      DESCRIPTION  1
DID  1              DOC  1              DOCUMENTED  1       DON  2
EACH  1             EDITOR  1           ENCLOSED  1         ENCLOSURE  1
EXCELLENT  1        EXPLANATION  1      EXPRESS  1          FILE  2
FINE  1             FOR  3              FORUM  1            FUNCTIONAL  1
FURTHER  1          HAMILL  1           HAVE  1             HELPFUL  1
HIXSON  1           HOWEVER  1          I  4                IN  1
IS  2               ISSUE  1            IT  1               JCP  4
JOB  1              JOURNAL  3          JULY  1             JUNE  1
LIKE  1             LISTING  2          MAJOR  1            MAY  2
MICRO  2            MJ  1               MURRAY  2           MY  1
O  1                OF 10               OFFERS  1           P  1
PASS  1             PETER  2            PRESENTING  1       PROCEDURE  4
READERS  1          REFER  1            REVIEW  1           ROAD  1
RUN  1              S  1                SAMPLE  1           SINCERELY  1
STAFF  1            STATEMENTS  2       STEP  1             TEXT  1
THANK  2            THAT  2             THE 11              THERE  1
THOUGHT  1          TN  1               TO  4               TX  1
TXT  1              USE  1              USED  1             VARIOUS  1
WANT  1             WHICH  1            WILL  1             WILLIAMS  1
WORDCNT  3          WOULD  2            YOU  2              YOUR  2

READY


FLEX

+++PDEL,1.TMP3.TXT


DELETE "1.TMP3.TXT"? Y

+++
...PROCEDURE COMPLETED

+++
```

Tel : Witham 512469
(STD Code 0376)

47, Collingwood Road,
Witham, Essex.
CM8 2DZ.

England.
16th November 1980

Mr. Don Williams,
'68' Micro Journal,
3018 Hamill Road,
PO Box 849,
Hixson, Tennessee 37343

Dear Don,

I have just started a subscription to '68', after borrowing some back issues from a friend. I was impressed with the programs linking BASIC to FLEX and EDITOR to FLEX.

As I have only the cassette Editor and not the disk version, I used Art Weller's program and wrote programs in the Flex transient command space to allow Save and Read Disk commands using Cassette Editor. I have not attempted to insert line numbers, since the Editor "RENUMBER" function does this perfectly well.
Since SAVE is a FLEX system command, I have used the names EDSAVE.CMD and EDREAD.CMD instead.

To call from Editor, the procedure is:
* DC EDREAD FILENAME (followed by) * RENUMBER
* DC EDSAVE FILENAME
I hope these routines may be of assistance to other readers.

Yours sincerely,

Bill Hughes

Bill Hughes

```
                * DISK SAVE FOR TSC CASSETTE EDITOR *
                * WRITTEN BY WILLIAM A.HUGHES        *
                * 47 COLLINGWOOD ROAD, WITHAM,       *
                * ESSEX, CM8 2DZ, ENGLAND.           *
                * EDSAVE
                * DEFINE FLEX2 ENTRY POINTS AND VARIABLES *
AD2D            GETFIL  EQU   $AD2D
AD33            SETEXT  EQU   $AD33
AD03            WARMS   EQU   $AD03
AD3F            RPTERR  EQU   $AD3F      ** USES "DC" COMMAND **
B406            FMS     EQU   $B406      ** OF ART WELLER     **
B403            FMSCLS  EQU   $B403
A840            FCB     EQU   $A840
0097            FILBEG  EQU   $97
0099            FILEND  EQU   $99
A100                    ORG   $A100
A100 20 03      EDSAVE  BRA   SAVE1
A102 04         VN      FCB   4
A103            XTEMP   RMB   2
A105 CE A8 40   SAVE1   LDX   #FCB
A108 BD AD 2D           JSR   GETFIL     GET FILE SPEC
A10B 25 46              BCS   ERROR      ANY ERRORS?
A10D 86 01              LDAA  #1         CODE FOR TEXT
A10F A7 00              STAA  0,X
A111 BD AD 33           JSR   SETEXT
A114 86 02              LDAA  #2         CODE FOR WRITE OPEN
A116 A7 00              STAA  0,X
A118 BD B4 06           JSR   FMS
A11B 26 36              BNE   ERROR
A11D DE 97              LDX   FILBEG     POINT TO START
A11F FF A1 03           STX   XTEMP
A122 9C 99      SAVE2   CPX   FILEND     FINISHED?
A124 27 1E              BEQ   CLOSE
A126 08                 INX              IGNORE LINE NO.
A127 08                 INX
A128 08                 INX
A129 A6 00      SAVE3   LDAA  0,X        POINT TO CHARACTER
A12B 9C 99              CPX   FILEND     FINISHED?
A12D 27 15              BEQ   CLOSE
A12F 08                 INX
A130 FF A1 03           STX   XTEMP
A133 CE A8 40           LDX   #FCB
A136 BD B4 06           JSR   FMS        WRITE CHAR TO DISK
A139 26 18              BNE   ERROR
A13B FE A1 03           LDX   XTEMP
A13E 81 0D              CMPA  #$D        IS CHAR C/RT
A140 27 E0              BEQ   SAVE2
A142 20 E5              BRA   SAVE3      GET NEXT CHAR
A144 CE A8 40   CLOSE   LDX   #FCB       CLOSE FILE
A147 86 04              LDAA  #4
A149 A7 00              STAA  0,X
A14B BD B4 06           JSR   FMS
A14E 26 03              BNE   ERROR
A150 7E AD 03           JMP   WARMS      DONE -- RETURN
A153 BD AD 3F   ERROR   JSR   RPTERR
A156 BD B4 03           JSR   FMSCLS     ERROR -- CLOSE FILE
A159 7E AD 03           JMP   WARMS        & RETURN
                        END   EDSAVE
   NO ERROR(S) DETECTED
```

```
                * DISK READ FOR TSC CASSETTE EDITOR *
                * WRITTEN BY WILLIAM A. HUGHES       *
                * 47 COLLINGWOOD ROAD, WITHAM,       *
                * ESSEX, CM8 2DZ, ENGLAND.           *
                * EDREAD
                * DEFINE FLEX2 ENTRY POINTS AND VARIABLES
AD2D            GETFIL  EQU   $AD2D
AD33            SETEXT  EQU   $AD33
AD03            WARMS   EQU   $AD03
AD3F            RPTERR  EQU   $AD3F      ** USES "DC" COMMAND **
B406            FMS     EQU   $B406      ** OF ART WELLER     **
B403            FMSCLS  EQU   $B403
A840            FCB     EQU   $A840
0099            FILEND  EQU   $99
A100                    ORG   $A100
A100 20 01      EDREAD  BRA   READ1
A102 06         VN      FCB   6
A103 CE A8 40   READ1   LDX   #FCB
A106 BD AD 2D           JSR   GETFIL     GET FILE SPEC
A109 25 2E              BCS   ERROR
A10B 86 01              LDAA  #1         CODE FOR TEXT / READ
A10D A7 00              STAA  0,X
A10F BD AD 33           JSR   SETEXT
A112 BD B4 06           JSR   FMS
A115 26 22              BNE   ERROR
A117 DE 99      READ2   LDX   FILEND
A119 6F 00              CLR   0,X        RESERVE SPACE
A11B 6F 01              CLR   1,X        FOR LINE NUMBER
A11D 6F 02              CLR   2,X
A11F 08                 INX              ADVANCE POINTER
A120 08                 INX
A121 08                 INX
A122 DF 99              STX   FILEND
A124 CE A8 40   READ3   LDX   #FCB
A127 BD B4 06           JSR   FMS        READ CHAR FROM DISK
A12A 26 0D              BNE   ERROR
A12C DE 99              LDX   FILEND
A12E A7 00              STAA  0,X
A130 08                 INX
A131 DF 99              STX   FILEND
A133 81 0D              CMPA  #$D        END OF LINE?
A135 27 E0              BEQ   READ2
A137 20 EB              BRA   READ3
A139 A6 01      ERROR   LDAA  1,X
A13B 81 08              CMPA  #8         EOF?
A13D 26 19              BNE   ERROR2
A13F 86 04              LDAA  #4         DONE -- CLOSE FILE
A141 A7 00              STAA  0,X
A143 BD B4 06           JSR   FMS
A146 26 0A              BNE   ERROR2
A148 DE 99              LDX   FILEND
A14A 09                 DEX              BACKTRACK POINTER
A14B 09                 DEX              TO END OF TEXT
A14C 09                 DEX
A14D DF 99              STX   FILEND
A14F 7E AD 03           JMP   WARMS
A152 BD AD 3F   ERROR2  JS    RPTERR
A155 BD B4 0            JSR   FMSCLS     ERROR -- CLOSE FILE
A158 7E AD 03           JMP   WARMS        & RETURN
                        END   EDREAD
   NO ERROR(S) DETECTED
```

BLUE HAT SOFTWARE COMPANY
Box 4127 Flint, Michigan 48504

Dear Sirs:

In reference to the review of DIXIE appearing in your Nov. '80 issue: Please inform your readers that a patch for Percom SUPER BASIC (supporting data files) and an MPX-to-DIXIE disk conversion utility are now available. Both pieces of software are included in the $60 price for DIXIE.

Sincerely,

Larry E. Preston

SEPTEMBER 12, 1980

MR. DAVID WEEKS
F & D ASSOCIATES
1210 TODD ROAD
NEW PLYMOUTH, OHIO 45654

DEAR MR. WEEKS:

JUST A NOTE OF APPECIATION FOR YOUR EXCELLENT PRODUCTS AND SERVICE. SEVERAL OF MY 6800 FRIENDS HAVE REPORTED SIMILAR EXPERIENCES WITH YOU AND YOUR FIRM. WE T ONLY WISH YOU THE BEST OF CONTIN ED SUCCESS, BUT ALSO WANT TO LET OTHER 6800 USERS KNOW OF OUR EXPERIENCE. TH NKS AGAIN FOR YOUR QUALITY PRODUCTS AND SERVICE.

Ernest Steve Watson

ERNEST STEVE WATSON

CC: '68' MICRO JOURNAL

9917 La Duke Drive
Kensington, Md 20795
30 September 1980

Don Williams Sr.
'68' Micro Journal
3018 Hamill Rd.
P.O. Box 849
Hixson, Tennessee 37343

Dear Mr. Williams:

I tried the modification described by William R. Hamblen in the April 1980 issue of '68' Micro Journal to provide a "home-up" function for a SWTP-CT-64 in the scrolling mode. It performed a "home-up" but did not clear the screen. In an attempt to provide a clear screen, I also connected pin 10 of J2 to pin 6 and the decoder. The result is erratic! sometimes the screen is cleared, sometimes a second pulse is needed. Increasing R47 to 10K and substituting LS chips for t e "erase-to-end-of-frame" circuits does not help. Can anyone suggest a solution ? Another modification I would like to make is to remove the extra line-feed when typing in BASIC while still ignoring line-feeds and using the RETURN key to perform the CR/LF function. Has any reader solved this problem ?

I find your journal both helpful and interesting. While clearly bucking a trend, I would welcome more attention to tape rather than disc systems. I am enjoying JPC's new BASIC/3 with 4800baud tape, even though there are still a few bugs JPC has promised to fix. My previous experience with the company has been so good that I am willing to hang in there. BASIC/3 has a host of good features like right-adjust and abbreviated commands, and is about 50% faster than SWTP v.2.3

Sincerely yours,

E.R. Schmerling

E.R. Schmerling

THIS VERSION OF FLEX tm PRINT.SYS DRIVES THE MODEL 8300 C. Itoh PRINTER WITH THE JPC PRODUCTS TC-3 CASSETTE INTERFACE BOARD.

```
                        NAM    PRINT.SYS
                *INTERFACE PARALLEL PRINTER
                *WITH JPC CASSETTE BOARD-PORT 7
         E01E   PIAB    EQU    $E01E      ($801E - 6800 FLEX)
                *INITIALIZE 8300 PRINTER
CCC0                    ORG    $CCC0      (ACC0 - 6800 FLEX)
CCC0 86  FF     PINIT   LDA    #$FF       TELL PDR WANT
CCC2 B7  E01E           STA    PIAB       ALL OUTPUT LINES.
CCC5 86  2E             LDA    #%00101110 CB2 AS STROBE &
CCC7 B7  E01F           STA    PIAB+1     CB1 ACK F AS b7.
CCCA 86  11             LDA    #$11       DC1 'SELECT' TO
CCCC B7  E01E           STA    PIAB       ENABLE 8300 PRINTER.
CCCF 39                 RTS
                *PRINTER READY?
CCD8                    ORG    $CCD8
CCD8 7D  E01F   PCHK    TST    PIAB+1     IS b7 SET ?
CCDB 39                 RTS
                *OUTPUT CHARACTER
CCE4                    ORG    $CCE4
CCE4 8D  F2     POUT    BSR    PCHK       WAIT FOR ACK
CCE6 2A  FC             BPL    POUT       TO SET b7.
CCE8 7D  E01E           TST    PIAB       CLEAR b7.
CCEB B7  E01E           STA    PIAB       SEND CHARACTER.
CCEE 39                 RTS

0 ERROR(S) DETECTED
```

DON'T LET YOUR TC-3 CASSETTE INTERFACE GATHER DUST WHEN YOU ARE RUNNING WITH A FLOPPY SYSTEM. IT IS VERY CONVENIENT TO USE THE B SIDE PIA OUTPUT FOR VARIOUS TASKS. RUN 11 WIRES TO A DB-25 CONNECTOR FOR QUICK CHANGES.

STANDARDIZE ANY PINOUT YOU PREFER. I USED DB-25 P14 - BIT 0 THROUGH P21 FOR BIT 7. C1 (CB1) MUST CONNECT TO P10 AT THE 8300 PRINTER; THAT IS THE ACKNOWLEDGE OUTPUT OF THE PRINTER. I WIRED P17 & P28 TOGETHER AT THE PRINTER PLUG AND CARRIED THAT GROUND OVER TO THE G PIN ON THE TC-3. C2 (CB2) MUST CONNECT TO P1 AT THE PRINTER; IT STROBES THE PRINTER THAT THE DATA IS VALID.

FOR SHORT RUNS IT IS PROBABLY NOT NEEDED, BUT I CONNECTED THE GROUND FOR EACH DATA BIT TO ALTERNATE WIRES IN THE RIBBON CABLE. THIS WAS DONE AT THE PRINTER PLUG ONLY AND THE DB-25 ENDS WERE LEFT FLOATING. (GROUND PATH WAS P17 & P28 AS MENTIONED ABOVE.) BIT 7 IS NOT REQUIRED BY TH 8300 PRINTER.

IN EARLIER VERSIONS I USED A SOFTWARE DELAY ROUTINE FOR THE INPUT PRIME SIGNAL. FOR MY APPLICATIONS IT DOES NOT SEEM TO BE REQUIRED.

R. PATTERSON - P.O. BOX 386 - MTN VIEW, AR 72560

HELP

NEED SWTPC MP-A OR MP-A2 CPU CARD OR SIMILIAR FOR SS-50 BUSS. PREFER TO HAVE CARD OPERATIONAL, BUT WILL CONSIDER CARD THAT NEEDS REPAIR IF NECESSARY. SEND INFO, CONDITION, PRICE, ETC TO:
GEORGE KELM - PO BOX 160 - YAP ISLAND, GUAM 96943.

***

HELP!! I FREQUENTLY USE THE TSC'S TEXT PROCESSOR OPERATING UNDER MINIFLEX TO SET TEXT FILES RUNNING HUNDREDS OF THOUSANDS OF CHARACTERS. WHO CAN SUGGEST A PRINT SYSTEM FOR USE WITH THIS THAT WOULD USE H2500 TO H6500 FOR A PRINT BUFFER INSTEAD OF A TINY BUFFER USED BY MINIFLEX? THE PRESENT BUFFER CAUSES A DISK READ FOR EACH TWO LINES PRINTED. THE WEAR ON THE DISK DRIVES IS UNBEARABLE.
JOHN P TUCKER - PO BOX 2898 - LAREDO, TEXAS 78041.

***

CLASSIFIED ADS

CT-64+9" MONITOR $175, AC-30 CASSETTE INTERFACE $30 CALL - J WYTSMA (404)262-3101.

***

4K SWPTC MEMORY BOARDS $35 EACH, PERCOM DISK CONTROLLER WITH 3 DRIVE CABLES, 6800 DOS & FLEXTRAM, 6809 MPX DOS, BOX 10 DISKETTE, APPROX $250 VALUE, WILL SELL FOR $140 PERCOM ELECTRIC WINDOW WITH 6800 WINDEX /9 ROMS $235 COMPLETE. 6800 FLEX SOFTWARE —BIG DISCOUNT!! CALL FOR PRICES.
DR. MATTHEW SCUDIERE - 100 CEDAR LANE - OAK RIDGE, TN 37830 (615)576-7648 (DAY) / 482-1355 (EVENING).

***

CENTRONICS P-1 W/MP-LA CABLE $250, SWPTC CT-64 $200, HITACHI 9" MONITOR $75, SWPTC AC-30 $75.
SHILLING - 2003 FAIRMEADOW - ARLINGTON, TX 76012 (817)461-2239.

***

6800/6809 PASCAL

DYNASOFT PASCAL is a cassette based PASCAL subset designed to run on most 6800/6809 systems with 12K or more of memory.

DYNASOFT PASCAL includes most of the control structures of standard PASCAL including IF-THEN-ELSE, CASE-OF-OTHERWISE, WHILE-DO, REPEAT-UNTIL, FOR-TO/DOWNTO-DO, and recursive PROCEDURE's and FUNCTION's. It supports the data types INTEGER, CHAR, BOOLEAN, scalar (user-defined), subrange, pointer and ARRAY. It is built around a one pass compiler which produces fast, compact p-code and comes complete with line-oriented text editor, p-code interpreter, and program SAVE and LOAD routines. The whole system resides in less than 8K.

LAST TIME OFFER AT THIS PRICE!

The cassette version with manual is priced at $35 plus $3 for postage and handling. Please specify 6800 or 6809.

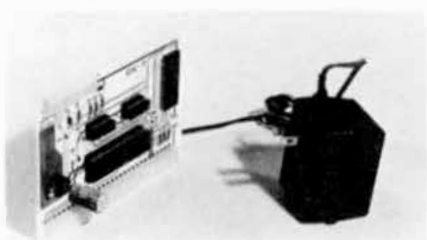
Dyna soft systems ltd.

P. O. BOX 51, WINDSOR JCT.
NOVA SCOTIA, CANADA
B0N 2V0 (902) 861-2202

Give your computer ... ..the time of day!

WITH THE CK - 7 CLOCK FOR SWTPC COMPUTERS.

- A TRUE CLOCK, not a timer, keeps time continuously without servicing by the computer. Provides hours, minutes, seconds.



- COMPLETE KIT - $59.95

★ power supply allows clock to run with computer power turned off.

phone
505 294-4623

Terms: Cash, MC or Visa

JPC PRODUCTS CO.

add $3 shipping

WE'VE GOT YOUR 68XX SOFTWARE

GYPSY TERMINAL COMMUNICATIONS PACKAGE

- Completely interrupt driven
- Send-Accept data files to-from host
- Your computer functions ON-LINE as intelligent terminal to host machine
- User-settable handshake. Most common protocols can be used
- Spool incoming data to printer
- SET command allows setting about 50 different parameters to user's desired configuration
- Baud rates up to 9600 permitted

Available NOW for TANO Outpost 11 and coming soon for SWTPC and WAVEMATE.

ED SMITH'S SOFTWARE WORKS

** 6800 Software
- Relocating Assembler
- Relocating Recursive Macro Assembler
- Disassembler--Source Generator
- Disassembler--Segmenting Source Generator
- Disassembler--Trace Debugger
- EPROM Support Relocator
- SMITHBUG Monitor
- 6800→6809 Cross Macro Assembler

** 6809 Software
- Relocating Assembler
- Relocating Recursive Macro Assembler
- Disassembler--Source Generator
- Disassembler--Segmenting Source Generator
- 6809→6800 Cross Macro Assembler

Available on FLEX 2.0, FLEX 9.0, SSB 5" or 8" disks.

OSBORNE BUSINESS PROGRAMS in TSC XBASIC

- Accounts Receivable
- Accounts Payable
- General Ledger

- We specialize in software for TANO, SWTPC, and WAVEMATE series 2000 computers
- Visa or Mastercard accepted
- Dealer Inquiries welcome

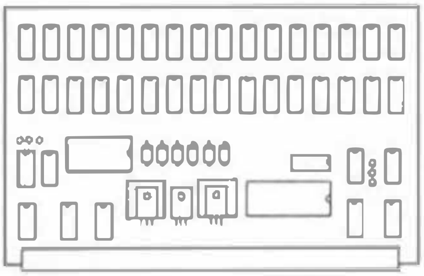
FLEX is a TradeMark of Technical Systems Consultants

Great Plains Computer Company, Inc.

P.O. Box 916, Idaho Falls, Idaho 83401 208—529-3210

64K DYNAMIC RAM

FOR THE SS-50C BUS



- DECODES 16 OR 20 ADDRESS-LINES
- 1 MHz OPERATION

- FULL KIT	US$499	CDN$579
- ASSEMBLED	US$599	CDN$699

- DELIVERY 4-6 WEEKS
- DEALER INQ. INVITED

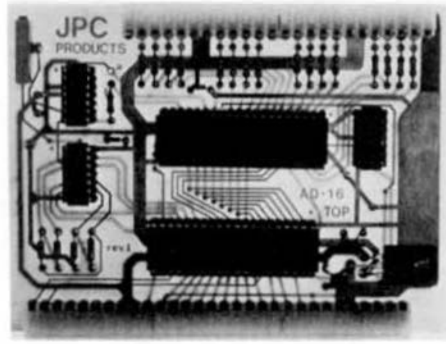
VISA

VISA, CERT. CHEQUE, MONEY ORDERS ACC.
ONTARIO RES. ADD 7% SALES TAX

RCE ROBERTS CONTROL EQUIPMENT
3840 WESTON RD UNIT 3 WESTON
ONTARIO CANADA M9L 1W2

CANADA: (416)749-5062
USA: (716)631-8178

JPC PRODUCTS FOR 6800 COMPUTERS



USES ONE I/O SLOT

16 CHANNEL A/D BOARD

- 8 BIT DATA
- SOFTWARE CONTROLLED GAIN
- 3300 SAMPLES PER SECOND
- ± 0.7% ACCURACY

COMPLETE KIT: AD-16 $69.95

Terms: Cash, MC or Visa; Shipping & Handling $3.00

JPC products

Order Phone (505) 294-4623
P.O. Box 5615
Albuquerque, N.M. 87185

THE SCREDITOR II AND TREK6864 ARE HERE!!

We've spent the last year working on it. We've taken your ideas and added some of our own, and come up with the most powerful, easy-to-use, EDITOR/FORMATTER available today for the 6800! Look at a few of the features of the SCREDITOR II and see why we say it's the BEST editor available for MEMORY-MAPPED DISPLAYS!

- FULL SCREEN FORMATTING - Tabs and margins are fully dynamic - set or move them anywhere, any time! Word wrap, paragraph splitting, line welding all honor the current margins!
- DYNAMIC SCREEN DISPLAY - What you do is immediately visible on the screen - inserts, deletes, line changes, copies, file merges, etc..., all happen as you watch!
- MULTI-MODE EDITING - In LINE mode, no wrap or paragraph filling takes place. Ideal for Basic, Pascal, Forth or assembly language coding! In TEXT mode, automatic word wrap, paragraph filling, etc..., make document preparation a breeze!
- REPLACEMENT STRINGS - Define common words, phrases, even command sequences as a single character - one keystroke entry replaces a lot of typing! Even save and load the replacements to and from disk!
- KEYBOARD QUEUE BUFFER - For systems which can support interrupt operation, full type-ahead is a standard feature!
- SYSGEN CUSTOMIZATION - No more hours of machine code patches with the SCREDITOR II - we provide a complete SYSGEN program which will do the work for you - simply answer the questions, and your patches are done!

The SCREDITOR II now supports 30 commands, 24 screen operators, and is completely upward-compatable with all text and source file handling programs! A full co-resident processor to be announced soon! Available now for TSC FLEX 1.0 and 2.0, SSB DOS68 and SWTPCo mini-FLEX, and all popular MEMORY-MAPPED DISPLAYS!

PROGRAM DISK, 100 PAGE MANUAL . . .	$79.95
MANUAL ONLY	$49.95
SOURCE LISTING	$39.95
SOURCE CODE ON DISK	$229.95

(This ad was composed using only transfer lettering and the new SCREDITOR II.)

VA. residents add 4% state sales tax. MC, VISA, COD's, personal checks accepted. Checks require longer to process. For orders under $100, add $5 for shipping and handling.

At last a REAL-TIME, LIVE-ACTION TREK-type simulation for the 6800 is here! TREK6864 is the most exciting arcade-type game ever available for 6800 users!

- AS YOU WATCH, the DARSTANG invaders implement confusing battle plans, their fleet moving toward you, firing as they come, their photon torpedoes homing on you!
- AS YOU WATCH, your energy decrements, becoming dangerously low and you cannot warp because of damage. But wait!
- AS YOU WATCH, a message flashes - WARP DRIVES REPAIRED! Now to find a base - you scan - a base is found! You rotate the CENTURIAN to prepare to warp...and a torpedo hits! CLOAKING DEVICE DAMAGED! You warp!
- AS YOU WATCH, the quadrant you entered is infested with more DARSTANG...and they have spotted you! Even as you move toward your base, you are hit again and again! IMPULSE ENGINES DAMAGED! CLOAKING DEVICE REPAIRED! You cloak quickly to await repairs, as the DARSTANG begin to sweep the quadrant in confusion! Will your energy run out before the repairs are complete? Will the DARSTANG destroy your base before you can dock? Only time will tell as you play TREK6864!

This REAL-TIME game is the most engrossing simulation you have ever seen for the 6800! If you have a 64x16 MEMORY-MAPPED board, and an ACIA or PIA keyboard port, you too can play TREK6864! And, like our SCREDITOR II, we provide a SYSGEN program to make modifications easier! (80 character versions available soon.) And, finally, the price is right!

TREK6864 PROGRAM DISK, MANUAL	$24.95
PROGRAM ASSEMBLY LISTING	$39.95
PROGRAM SOURCE CODE ON DISK	$99.95

FLEX is a TradeMark of Technical Systems Consultants

Alford and Associates
P. O. Box 6743
Richmond, Va., 23230
804-320-6722

SMOKE SIGNAL BROADCASTING

Presents

3 Powerful New SS-50/SS-50C Boards

DCB-4
Disk Master
Double Density Controller Board and DOS68D Double Density DOS
$449.00

The new DCB-4 is a truly state-of-the-art development which allows up to 366K bytes to be stored on a single 5¼" disk and has these outstanding features:

- Up to four 5¼"and four 8" drives can be handled in the same system with a user definable logical unit table. (DOS68D will be compatible with future hard disk systems).
- Under software control, the user can select the following for any drive:
  - ☆ Single sided or double sided operation.
  - ☆ Single density or double density data.
  - ☆ 5¼" or 8".
  - ☆ Stepping Rate.
  - ☆ 40 track or 35 track density on double sided 5¼" drives.
  - ☆ User can select the system boot configuration.
- Occupies only 16 bytes of memory space (F760-F76F standard). User selectable to any 16 byte address space.
- Can read and write a single sector by itself. On-board buffer memory allows full interrupt capability in interrupt driven systems. Once data transfer has been initiated, no more processor time is required.
- Contains extended decoding circuitry for extended addressing per SS–50C bus which can be enabled by an option jumper.
- SSB provides a means for copying software written by older versions of DOS68 to be read by DOS68D. All new media formatted by DOS68D can be read by all older versions of DOS68. DOS68 is SSB's 6800 disk operating system.
- Track Ø of side Ø is recorded in single density per IBM standard.
- Phase-locked-loop assures highest data integrity attainable.

All of these features are available for immediate delivery on one standard 5½" x 9" 50 pin SS–50/SS–50C card for only $449.00. The price includes DOS68D version 5.1, MONITOR object code on diskette, and a manual with the source listing.

SCB-69
Super Computer Board
6809CPU Board
$299.00

The most versatile 6809 CPU Board on the market is now available from Smoke Signal Broadcasting and has the following features:

- Standard 2 MHz operation. (Shipping 1.5 MHz until 68B09 available)
- 20 bit address generation for up to 1 Mbyte of memory. Uses an improved address translation RAM which is compatible with present extended addressing schemes yet requires much less overhead when used in multi-user systems.
- All on-board devices can be switch selected to occupy any or all extended pages. Any on-board device may be disabled and its memory space is then available for external memory.
- Standard real–time clock (time–of–day, day–of–week, day–of–month) with battery back up capable of generating programmable interrupts.
- Up to 20K of EPROM can be installed on the CPU Board.
- Standard 1K of RAM on board.
- Includes improved 6809 Monitor (and source listing).
- Contains an FPLA for decoding EPROM address and optional devices. Switches are used to select 2K/4K EPROM and Fast/Slow I/O.
- Contains provision for optional 9511/9512 floating point processor.
- NMI line is user selectable to work with either SS–50 or SS–50C busses.

Price for the new SCB–69 is only $299.00 for an assembled, burned-in fully tested board.

M-32-X
32K
Memory Board
$539.00 $439.00

The first and only 32K Static Ram Board on standard size (5½" x 9") SS–50/SS–50C Bus Circuit Card is made by Smoke Signal.

- Switch selectable to any 4K boundary.
- Any 4K block may be switch enabled or disabled.
- Fully compatible with SS–50C extended addressing (allows memory decoding up to 1 Mbyte).
- Extended addressing capability may be switched off for compatibility with SS–50 systems.
- Gold Bus Connectors for high reliability.
- Guaranteed 2MHz operation (tested at 2.2 MHz).
- Low power consumption – 8 volts at 2.4 amps typical.

M–32–X 32K Memory Board is priced at $539.00.

M–24–X 24K Memory Board expandable to 32K, is $439.00.

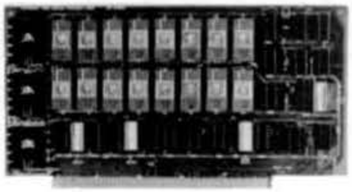
And our M–16–X 16K board is back to the old price of $299.00.

SMOKE SIGNAL BROADCASTING®

31336 Via Colinas, Westlake Village, CA 91361, (213) 889-9340

DIGITAL RESEARCH COMPUTERS
(214) 271-3538

32K S-100 EPROM CARD
NEW!



$74.95 KIT

USES 2716's

Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $14.95 EA. With Above Kit.

KIT FEATURES
1. Uses +5V only 2716 (2Kx8) EPROM's
2. Allows up to 32K of softwa e on line!
3. IEEE S-100 Compatible
4. Addressable as two independent 16K blocks
5. Cromemco extended or Northstar bank select
6. On board wait state circuitry if needed
7. Any or all EPROM locations can be disabled
8. Double sided PC board, solder-masked, silk-screened
9. Gold plated contact fingers
10. Unselected EPROM's automatically powered down for low power
11. Fully buffered and bypassed
12. Eas and quick to assemble

16K DYNAMIC RAM PARTIALS
LOOK! INTEL 2108 8K X 1 RAMS LOOK!
8 FOR $9.95 32 FOR $35
FACTORY PRIME!

Huge special purchase of INTEL Dynamic RAM's. These are 2108-4, 300NS, 8K, Ceramic DIP. The 2108 is the INTEL 2116 (16K) tested for either upper or lower 8K only. These are factory prime. Full Spec. See INTEL 1978 Cat. for details or Memory Design Handbook for application data. Both IMSAI and EXTENSYS did mfg. S-100 RAM boards using these devices. — P.S. These devices will not work in the SD EPANDORAM™. Please specify upper or lower 8K. (S1626 or S1627). A super easy RAM to interface to a Z80, 16 PIN DIP.

FOR 4MHZ PRICE CUT! **LOW POWER - 300NS** 8 FOR
2114 RAM SALE! $37.50

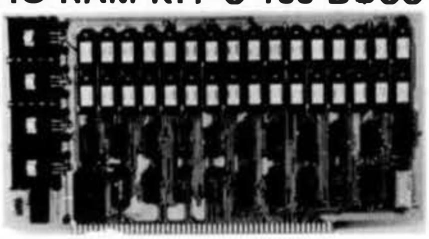
4K STATIC RAMS, MAJOR BRAND, NEW PARTS.
These are the most sought after 2114's, LOW POWER and 300NS FAST.
8 FOR $37.50

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!

$199.95 KIT

FOR 4MHZ ADD $10



KIT FEATURES
1. Addressable as four separate 4K Blocks
2. ON BOARD BANK SELECT circuitry (Cromemco Standard!) Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POW R under 1.5 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-$33
LOW PROFILE SOCKET SET-$12
SUPPORT IC'S & CAPS-$19.95
ASSEMBLED & TESTED-ADD $35

OUR #1 SELLING RAM BOARD!

16K STATIC RAM SS-50 BUSS

PRICE CUT!

$210 KIT

FULLY STATIC!

FOR 2MHZ ADD $10

FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES
1. Ad ressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Fully Bypassed
4. Double sided PC Board. Solder mask and silk screened layout
5. All Parts and Sockets included
6. Low Power: Under 1.5 Amps Typical

BLANK PC BOARD—$30 COMPLETE SOCKET SET—$12
SUPPORT IC'S AND CAPS—$19.95

STEREO!
NEW! **S-100 SOUND COMPUTER BOARD** NEW!

At last, an S-100 Board that unleashes the full power of two unbelievable General Instruments AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games or any other program. Sounds can be called in BASIC, ASSEMBLY LANGUAGE, etc.

COMPLETE KIT! **$84.95** (WITH DATA MANUAL)

BLANK PC BOARD W/DATA $31

KIT FEATURES:
- TWO GI SOUND COMPUTER IC'S.
- FOUR PARALLEL I/O PORTS ON BOARD.
- USES ON BOARD AUDIO AMPS OR YOUR STEREO.
- ON BOARD PROTO TYPING AREA.
- ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED.
- PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS.
- EASY, QUICK, AND FUN TO BUILD, WITH FULL INSTRUCTIONS.
- USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY

Both Basic and Assembly Language Programming examples are included.

SOFTWARE:

SCL™ is now available! Our Sound Command Language makes writing Sound Effects programs a SNAP! SCL™ also includes routines for Register Examine-Modify, Memory-Examine-Modify, and Play-Memory. SCL™ is available on CP/M* compatible diskette of 2708 or 2716. Diskette - $24.95 2708 - $19.95 2716 - $29.95 Diskette includes the source. EPROM'S are ORG at E000H.

4K DYNAMIC RAM BLOWOUT!
SAME AS INTEL 2107B!
4K RAMS AT AN UNBELIEVABLE 50¢ EACH!!!

Prime, new, National Semi., 1979 date coded, full spec. parts. N.S. #MM5280-5N. Same as INTEL 2107B-4, T.I. TMS4060, NECuPD411, etc. We bought a HUGE QTY. from a West Coast Distributor at truly DISTRESS PRICES! One of the most popular and reliable RAM's ever made. These parts have been used by almost all Major Computer Main Frame Mfg. the world over! Arranged as 4K x 1, 270 NS Access Time, 22 Pin Dip. These units DO NOT use multiplexed addressing, thus making REFRESH and other timing very simple. See INTEL MEMORY DESIGN HANDBOOK for full application notes. The NAT. SEMI. MEMORY DATA BOOK is available at most Radio Shack Stores. Prime units in original factory tubes!

#5280-5N 4096 BITS x 1 270 NS ACCESS
(With Pin Out Data) 8 FOR $4.95 32 FOR $16
FACTORY CASE (450 PCS) — $180
Sockets Special: 22 Pin Low Profile (With Purchase of 5280's) 8 FOR $1.

COMPUTER PARTS SPECIALS

74LS175 - .99	8035 Intel Single Chip CPU - 5.95
74LS240 - 1.79	Signetics 2901 4 Bit Slice - 6.95
74LS241 - 1.79	AMD 2903 4 Bit Super Slice - 12.50
74LS244 - 1.79	AMD 29705 Dual Port RAM - 8.95
74LS373 - 1.99	

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains 3 Tone Channels, Noise Generator, 3 Channels of Amplitu e Control, 16 bit Envelope Period Control, 2-8 Bit Parallel I/O, 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy interface to the S-100 or other busses.

SPECIAL OFFER: ~~$14.95~~ each $11.95 PRICE CUT! Add $3 for 60 page Data Manual.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $1.50 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tax. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. 90 Day Money Back Guarantee on all Items. Orders over $50, add 85¢ for insurance.

*TRADEMARK OF DIGITAL RESEARCH. NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

COMPUTER SYSTEMS CONSULTANTS, INC.

1454 LATTA LANE, CONYERS, GA. 30207
TELEPHONE 404-483-1717 OR 483-4570

SOFTWARE DEPARTMENT

680X DISASSEMBLER SYSTEM ($50.00 til Jan 30)	$100.00
—runs on 6800/1/9, analyzes 6800/1/5/9, 6502 codes	
—includes FLEX or SMOKE source programs	
—generates 6800/1 ASM code from 6800/1 code	
—generates 6809 ASM code from 6800/1/9 code	
—generates 6805 ASM code and 6502 ASM code	
—optionally generates 6809 position-independent code	
—automatic labels, optional FCB, FCC, FDB, RMB's	
—disk-to-disk or console, commands from menu or disk	
—generates FLEX or SMOKE and user-defined names	
—includes 680X ASM language XREF program	
680X ASSEMBLER LANGUAGE XREF PROGRAM	$ 25.00
—alphabetic x-reference program for any 6809 ASM code	
6805/6800/6802 MACROS FOR TSC 6809 ASM EACH	$ 50.00
—macro sets for cross-assembly on 6809 ALL 3	$100.00
FULL-SCREEN FORMS DISPLAY FOR TSC X-BASIC	$ 25.00
—display and edit for terminals and video displays	
—requires 5 K bytes on 6800 or 6809	
—Interactive forms generator/documentor provided	
TSC BASIC RESEQUENCING AND XREF PROGRAMS	$ 25.00
—processes TSC BASIC, X-BASIC, PC, XPC programs	
—partial and blank-resequence capabilities	
—alphabetized xref of all variables and BASIC verbs	
TSC X-BASIC DISK SORT/MERGE GENERATOR	$ 25.00
—interactively generates TSC XPC BASIC programs or subroutines for disk sort/merge	

(SPECIFY 5" OR 8", FLEX OR SMOKE DISK FOR ALL PROGRAMS)

HARDWARE DEPARTMENT

I/O SELECTRIC INTERFACE BOARD	$ 35.00
—ASCII (serial or parallel) in, 2 -50v solenoids out	
—transparent serial interface (RS-232/TTL + CTS)	
—up to 8 balls in 2708 PROM (BCD/Correspondence)	
SS-50 WIRE-WRAP BOARD (52-16 PIN EQUIVALENT)	$ 25.00
SS-30 WIRE-WRAP BOARD (32-16 PIN EQUIVALENT)	$ 12.50
SS-50 EXTENDER/TERMINATOR/SCOPE-PROBE BOARD	$ 30.00
—pullup SIP's for all signal lines included	
SS-30 SERIAL INTERFACE BOARD (1 ACIA)	$ 10.00
SS-50 FRONT PANEL DISPLAY BOARD	$ 10.00
—16 LED's display first digit of address	
STEEL DISK CABINET (2-MINIFLOPPIES+POWER SUPPLY)	$ 40.00

NO CREDIT CARDS — COD CHARGES EXTRA — US FUNDS ONLY
POSTAGE/HANDLING 5% US 1 % FOREIGN

DYNAMITE™

"THE CODE BUSTER"

DISASSEMBLES 6800 & 6809 MACHINE CODE INTO BEAUTIFUL SOURCE

- Convert your 6800 programs to 6809!
- Automatic LABEL generation
- Allows specifying FCB's, FCC's, FDB's, etc.
- Constants input from DISK or CONSOLE
- Automatically uses system variable NAMES
- DISK-to-DISK or DISK-to-CONSOLE operation

Includes 5" or 8" FLEX 9 diskette with relocatable object code. Full operating instructions (you'll learn in minutes!)

Order your DYNAMITE™ today
Only $60.00 postpaid in U.S. MC & VISA accepted
6809, FLEX 9, and 24K total RAM required
order from:

COMPUTER SYSTEMS CENTER

13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020

we also stock SWTPC, TSC, JPC products
hours 12-9 daily, 10-5 Saturday

Dealer inquiries welcome
FLEX is a trademark of TSC (Bless their hearts)

6809!

INTRODUCING THE NEW STATE-OF-THE-ART IN MICROCOMPUTER SOFTWARE FROM MICROWARE

OS9-1 SINGLE USER

OS9-1 WITH TAPE FILE MANAGER		
	on 2716's	$ 95.00
	on 2708's	$ 95.00
Manual & Source	only	$ 85.00
OS9-1 WITH DISK FILE MANAGER		
	on 2716's	$150.00
	on 2708's	$150.00
Manual & Source	only	$150.00
DEBUGGER PACKAGE (aprox 1K)		
	on 2716's	$ 50.00
	on 2708's	$ 50.00
	on tape	$ 35.00
	on disk	$ 35.00
Manual & Source	only	$ 50.00
INTERACTIVE EDITOR/ASSEMBLER		
	on 2716's	$180.00
	on 2708's	$180.00
	on tape	$150.00
	on disk	$150.00
Manual & Source	only	$150.00

Above items available after aprox. June 1, 1980.

SEE GIMIX AD
PAGES 3 & 48

COMING SOON!!!
BASIC09
OS9-2 MULTIUSER

When ordering, you must specify; type of CPU card, type of disk controller, size of media and starting address for your I/O ports.

From the company that puts it all together. GIMIX, SMOKE, SWTPC, MICROWARE, ANADEX, SPINWRITER, DIGITUS, HI-PLOT, MICROWORKS. . . .

H H H ENTERPRISES

BOX 493, Laurel, MD.
ZIP 20810
PHONE 301-953-1155

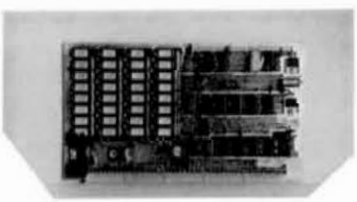
Model EP-2A-79
EPROM Programmer

Software available for F-8, 6800, 8085, 8080, Z-80, 6502, 1802, 2650, 6809 based systems.
EPROM type is selected by a personality module which plugs into the front of the programmer. Power requirements are 115 VAC 50/60 Hz, at 15 watts. It is supplied with a 36-inch ribbon cable for connecting to microcomputer. Requires 1½ I/O ports. Priced at $169.00 with one set of software. (Additional software on disk and cassette for various systems.) Personality modules are shown below.

Part No.	Programs	Price
PM 0	TMS 2708	$17.00
PM 1	2704, 2708	17.00
PM 2	2732	33.00
PM 3	TMS 2716	17.00
PM 4	TMS 2532	33.00
PM 5	TMS 2516, 2716, 2758	17.00
PM 8	MCM68764	35.00

Optimal Technology, Inc.
Blue Wood 127, Earlysville, Virginia 22936
Phone (804) 973-5482

SOUTHWEST MICROSYSTEMS
BRING YOUR SS-50 BUS OR EXORCISOR™ BUS 6800/09 COMPUTER INTO THE 1980's!!



FINALLY!! A 64K/256K MEMORY BOARD FOR THE SWTPC, MSI, GIMIX, D2, EXORCISOR™ & MICROCHROMA 68 COMPUTERS

- Operates with 6800, 6802, or 6809 processors.
- Low power consumption.

PRICE $385

Southwest Microsystems
Austin, Texas 78723
(512) 473-6983

Soft-R ware™
ease yourself into a new dimension of efficiency
MICRO UTILITY PACKAGES

FLEX(tm) / SSB — 6800 / 6809
Use Soft-R Ware(tm) to tame some of the tigers that routinely plague programmers and make the tigers purr like kittens. Easy and simple to use utilities run in fast machine language.

6800		6809
$29.95	ALPHABETIZING PACKAGE	$39.95 SSB only
$34.95	BASCOMPARE(TM)	$49.95
$34.95	DISK MAINTENANCE PACKAGE	$49.95 SSB only
$39.00	MICRO PROCEDURES(TM)	—
$24.95	RE-MAT/IT(TM)	— SSB only

See previous ads for descriptions.

MICRO Utilities provide easy flexibility to meet personal or customer needs for cutomized software. Simple language structures to enable a novice to build programs. The professional will appreciate the power of mainframe software at a fraction of its cost. Ideal for the systems house requiring frequent customization for a wide customer base, with the speed of machine language processing. Extensive documentation.

6800 or 6809
$139.00 MICRO SORTER©
$249.00 MICRO WRITER©
See previous ads for descriptions.

Numerous accounting, church, and professional packages also available. Menu driven for ease of operation with logical work flow procedures, all in RANDOM BASIC.

Specify 6800, 6809, FLEX©, SSB, 5" or 8" disks when ordering. Please add $2.50 for orders under $50.00 for shipping and handling.

Write call for more information.

Soft-R ware is marketed exclusively by
RIPLEY COMPUTERS
126 N. MAIN ST., SOUDERTON, PA 18964
(215) 723-1509

CHECK, MASTER CHARGE, VISA AND COD ACCEPTED
DEALER INQUIRIES INVITED

SOFT-R WARE, BASCOMPARE, POWER PATCH F-1 ARE TRADEMARKS OF RIPLEY COMPUTERS
MICRO WRITER, MICRO SORTER AND MICRO PROCEDURES ARE TRADEMARKS OF AUTOMOTION, INC.
*FLEX is a trademark of TECHNICAL SYSTEMS CONSULTANTS

Southeastern Micro Systems

NEW PRODUCT

SS-50 6800-6809 SS-50C 64·K MEMORY BOARD

kit or assembled

1981

6Axx CPUs & Support Chips

6800	$ 9.95
6802	13.95
6808	9.95
6809	19.95
6821	5.00
6850	4.50
6875	7.95
MC14411	9.95
MC1488	.95
MC1489	.95
8T95-98	1.50
8T26	1.75
8T28	1.75
MCM6575-A	10.00
74LS240	1.50
74LS241	1.50
74LS242	1.50
74LS243	1.50
74LS244	1.50
74LS245	2.25

•••••••EPROMS•••••

2708	8.95
2758	11.95
2716	15.95
2532	34.95

NEXT MONTH

SMS-30 BOARD

- FOR SS-50 & SS-50C
- 2 ACIA PORTS (addressable)
- 2 PIA PORTS ASB(addressable)
- ON BOARD BAUD RATE GEN.
- 1771 DISK CONTROLLER
- DATA SEPARATOR

- USES LOW COST 4116 DYNAMIC RAM
- TOTAL TRANSPARENT REFRESH: OPERATES LIKE STATIC
- LOW POWER CONSUMPTION 1AMP @ +5 500ma @+12 & -5 VDC
- ADDRESSABLE TO ANY 8-K BLOCKS
- CAN BE USED WITH 16K to 64K
- EXTENDED ADDRESSING TO 16-BOARDS
- USES MC3242 ADDRESS MULTIPLEXER AND MC3480 MEMORY CONTROLLER
- OPERATES ON SS-50 AND SS-50C BUSS
- SOLDER MASK AND SILKSCREENED BOARDS
- ALL MEMORY CHIPS ARE PRIME 4116-250ns
- KITS AND BOARDS ARE IN STOCK

••

µ-SYS™

64·K DYNAMIC

PRICES

- KIT WITH 16K RAM.............$ 219.00
- KIT WITH 64K RAM............. 345.00
- ASM. & TESTED WITH 16K RAM... 269.00
- ASM. & TESTED WITH 64K RAM... 395.00

SEND YOUR NAME AND ADDRESS FOR OUR CATALOG

404·922·1620

Charge to your MC/VISA

SOME DELAY FOR PERSONAL CHECKS

SEND YOUR ORDER TO 1080 IRIS DR P.O. BOX 293 CONYERS, GA. 30207

ADD $3.50 FOR SHIPPING IN USA $8.50 FOREIGN

BLITZ

SCREEN EDITOR FOR THE CT-82

- IDEAL FOR WORD PROCESSING OR PROGRAMMING
- THERE IS NO FASTER / EASIER WAY TO EDIT TEXT
- IT ALL HAPPENS IMMEDIATELY ON THE SCREEN SO YOU SEE EXACTLY WHAT YOU ARE DOING: INSERT CHARACTER, DELETE CHARACTER, INSERT LINE, DELETE LINE, SCROLL UP, SCROLL DOWN
- RUNS ON 6800 OR 6809 UNDER TSC's FLEX
- AVAILABLE ON 5 OR 8 INCH DISKETTE
- BEST OF ALL — YOU CAN BUY THE ENTIRE ASSEMBLY LANGUAGE SOURCE CODE, SO YOU CAN ADD YOUR OWN CUSTOM FEATURES
- FROM THE COMPANY THAT BROUGHT YOU THE MICROPI 4-USER PILOT/BASIC/EDITOR PACKAGE

$50 — OBJECT ONLY

$100 — SOURCE AND OBJECT

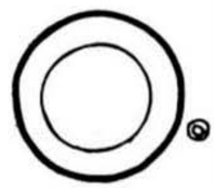
MICROPI
2445 N. Nugent
Lummi Island, Washington 98262

206-734-8248

How much will it cost to

LSI

**SIM68
$34.95**
plus shipping



LSI Enterprises Ltd.
P. O. 1227
Woodhaven, NY 11421
212-631-9242

VISA and Master Card accepted

upgrade your software to run on your 6809?

Run your 6800 software on your 6809 without cross assembly or modification.

Simulates all 6800 opcodes!

Available on FLEX(r) or Percom disk & KC tape.
(minifloppy only)
(r) - registered tm of TSC
US Shipping - $1; Foreign varies

HEMENWAY ASSOCIATES
SOFTWARE SOURCE BOOKS™

Your 6800 is up and running with HEMENWAY ASSOCIATES' complete software system. Software Source Books provide a powerful yet extensible programming package for business, scientific, or personal uses.

Combining detailed descriptions with COMPLETE SOURCE CODE LISTINGS, these books explain the internal operations and algorithms used in HEMENWAY ASSOCIATES' popular systems software.

Imagine getting a complete 6600 software library, and at these suprisingly low prices.

Remember, these are not just books; they are Software Source Books™ complete software resources! Order them today; VISA and MASTERCHARGE accepted.

CP/68 OPERATING SYSTEM

The most powerful operating system available for the 6800 family of microprocessors, this disk-based system features great flexibility. The user can add commands for special purposes. A single transient Peripheral Interchange Program (PIP) transfers data between devices. The system is relocatable anywhere in memory and fits in less than 8K. Other features include device-independent I/O and dynamic file allocation.

U.S. $34.95
Int. $52.45

XA6809
MACRO LINKING CROSS-ASSEMBLER

This new two-pass program generates relocatable and linkable code (requires LINK68). Resident on any 6800 system, XA6809 lets you produce code for a 6809 right now. This assembler has full macro facilities and features a COMMON section for the production of ROMable code, Conditional Assembly and fast execution.

U.S. $24.95
Int. $31.50

LINK68 LINKING LOADER

This is a one-pass linking loader which allows separately translated relocatable object modules to be loaded and linked together to form a single executable load module, and to relocate modules in memory. It produces a load map and a load module in Motorola MIKBUG loader format. This book provides everything necessary for learning about this system and the nature of linking loader design in general.

U.S. $7.95
Int. $11.95

RA6800ML RELOCATABLE
MACRO ASSEMBLER

This two-pass assembler produces a program listing, a sorted symbol table listing, and relocatable object code. The object code is loaded and linked with other assembled modules using LINK68. This book fully describes the 6800 assembly language and all major routines used, and includes flow charts, details on interfacing the assembler. Cross-referenced, showing all calling and called-by routines, pointers, flags and temporary variables.

U.S. $24.95
Int. $37.45

Structured BASIC language
STRUBAL+™COMPILER

The compiler features variable precision from 4 to 14 digits for business or scientific uses and Structured Programming forms. It produces Relocatable and linkable code. You can create data structures with mixed data types, COMMON and DUMMY sections. STRUBAL+™ includes a complete scientific package. It allows for string-handling and is extensible.

U.S. $49.95
Int. $74.95

Please send the following books

_____ copies CP/68 OPERATING SYSTEM
_____ copies STRUBAL+™ COMPILER
_____ copies XA6809 CROSS ASSEMBLER
_____ copies RA6800ML MACRO ASSEMBLER
_____ copies LINK68 LINKING LOADER

HEMENWAY ASSOC., INC.
101 TREMONT STREET
BOSTON, MA 02108
(617)426-1931

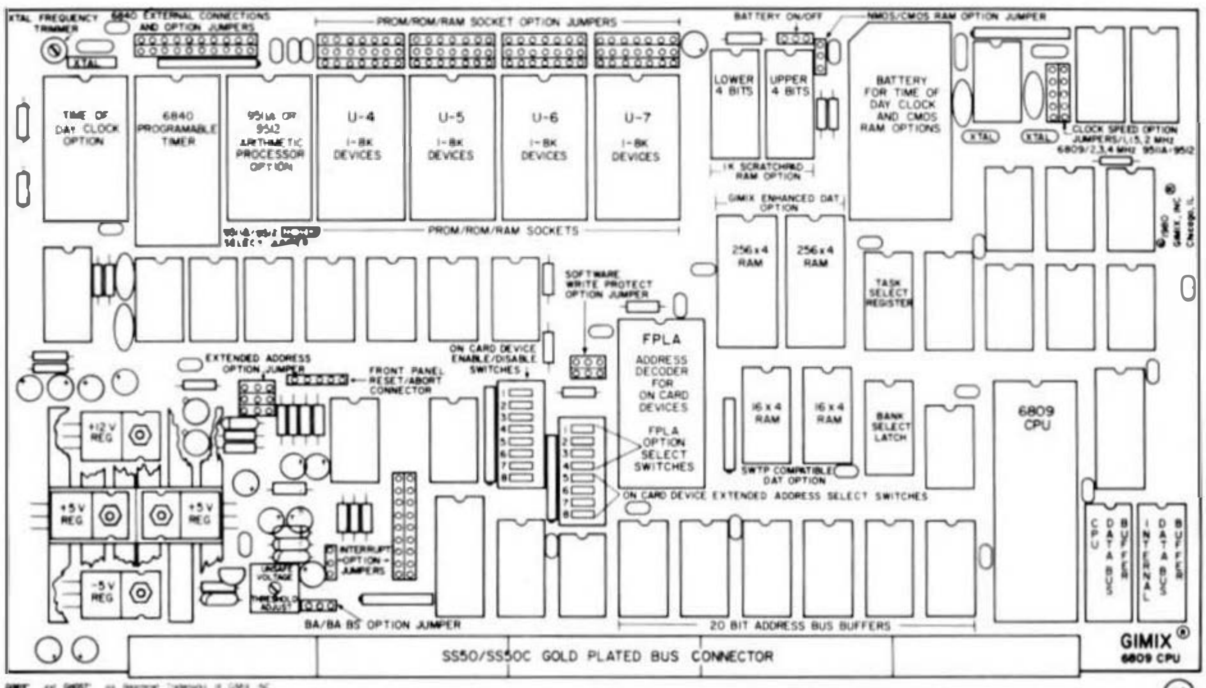
FOR NORTH AMERICA, ADD $.75/BOOK POSTAGE & HANDLING or $1.50/BOOK FOR FIRST CLASS
ALL OTHER DESTINATIONS, ADD $2.00/BOOK POSTAGE & HANDLING or $3.50/BOOK FOR PRIORITY MAIL

Name _____ Title _____ Company _____

Street _____ City _____ State/Country _____ Postal Code _____

☐ CHECK ENCLOSED in THE AMOUNT OF $
☐ BILL VISA ☐ BILL MASTERCHARGE
CARD NUMBER _____ EXP. DATE _____
☐ FIRST CLASS/PRIORITY ☐ BOOK RATE

GIMIX PROUDLY PRESENTS!!!



GIMIX 6809+ FUNCTION LAYOUT

SS50C 6809 CPU

The 6809 CPU card will be available in a standard version and our **6809 PLUS** version that is fully socketed to allow adding options at anytime.

- \+ A 6840 timer package that provides 3 independent 16 bit counters is included on all **6809 PLUS** cards.
- \+ A 9511 or 9512 Arithmetic Processors option with its own independent crystal that allows you to use 2, 3, or 4 MHz parts in any combination with the 6809 running at 1, 1.5, or 2 MHZ.
- \+ 1K of scratchpad RAM
- \+ A Time of Day Clock option with battery back-up. With this option you can also substitute 1K of CMOS RAM that will also be battery backed up.
- \+ User selectable processor speeds without having to change the crystal.

32K of PROM, ROM or RAM. Both versions have 4 sockets that can each hold from 1K to 8K parts. Single or multiple voltage parts can be used on the **PLUS** version. The standard version only allows the use of single voltage parts.

All on board devices and options can use extended addressing so that they will only respond to that page to which they are set.

The card is double buffered and allows versatility in the use of software and memory address control disciplines.

Please note that this card does not have an on board baud rate generator, and must be used in systems where baud rates (if needed) are provided elsewhere in the system.

GIMIX INC. 1337 WEST 37th PLACE • CHICAGO, IL 60609 • (312) 927-5510 • TWX 910-221-4055

'68' Micro Journal
3018 Hamill Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

000422 A/E
MR. MICKEY FERGUSON
BOX 708
TRENTON
GA 30752
MJ

3

SOFTWARE FOR THE HARDCORE

We know you hardcore bit hackers will recognize the computing power derived from combining the FORTH language with the 6809, today's most advanced 8 bit microprocessor.

And we know you'll understand this machine's 16 bit math, indirect addressing and two stacks are ideally suited for implementing FORTH.

But...should anyone need further convincing that FORTH provides a new dimension in power, speed and ease of operation, consider the following:

- It's a modern, modular, structured-programming high-level compiled language.
- It's a combined interpreter, compiler, and operating system.
- It permits assembler code level control of machine, runs near speed of assembler code, and uses less memory space than assembler code.
- It increases programmer productivity and reduces memory hardware requirements.
- It replaces subroutines by individual words and related groups of words called Vocabularies. These are quickly modified and tested by editing 1024-character text blocks, called screens, using built-in editor.

tFORTH is a basic system implemented for SS-50 buss 6809 systems with the TSC FLEX 9.0 disk operating system. It is available on 5¼" or 8" single density soft-sectored floppy disks. **$100.00**

tFORTH + consists of tFORTH plus a complement of the following FORTH source code vocabularies: full assembler, cursor controlled screen editor, case statements, extended data types, general I/O drivers. **$250.00**

firmFORTH is an applications package for use with tFORTH. It provides for recompilation of the tFORTH nucleus, deletion of superfluous code and production of fully rommable code. **$350.00**

Call or write today.

Also available for 6800

KENYON MICROSYSTEMS

3350 Walnut Bend • Houston, Texas 77042 • (713) 978-6933